Table of Contents

I - SVM (Support Vector Machine) ——— 1

II - Bayesian Decision Theory ——— 45

III - Maximum Likelihood / Maximum Entropy Duality - 91

IV - Expectation-Maximization Algorithm: Maximum Likelihood over Mixture of Distributions ——— 127

V - Spectral Analysis I: PCA ——— 161

I - SVM (Support Vector Machine)

A support vector machine is a supervised learning model (a model constructed from labelled examples - for each feature vector $\{x_i\}$ of the examples there is a label y_i that is known) for binary classification (the label y_i corresponds to one of two possible classes).

In this learning model, the binary labels may be either +1 or -1. A decision boundary corresponding to a hyperplane is defined with the objective of having the +1 label points falling on one side and the -1 ones falling on the other.

The hyperplane is defined as the set of points

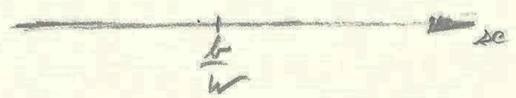
$$\mathcal{D} = \left\{ \{x\} \in \mathbb{R}^D : \{w\}^T \cdot \{x\} - b = 0 \right\} \quad (1)$$

where D is the dimension of $\{x\}$, and $\{w\}$ and b are parameters (weights, and bias, respectively).

For $D=1$, 2 and 3, the hyperplane has the representations shown below.

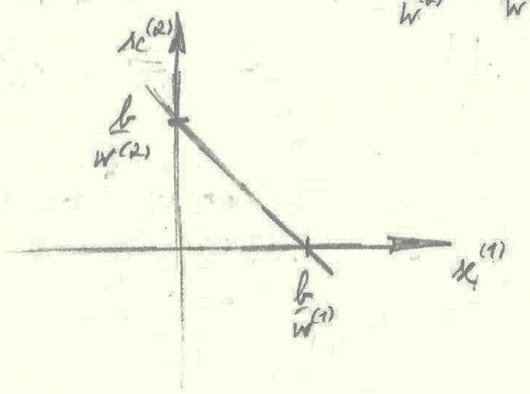
$D=1$:

$$wx - b = 0 \Rightarrow x = \frac{b}{w}$$



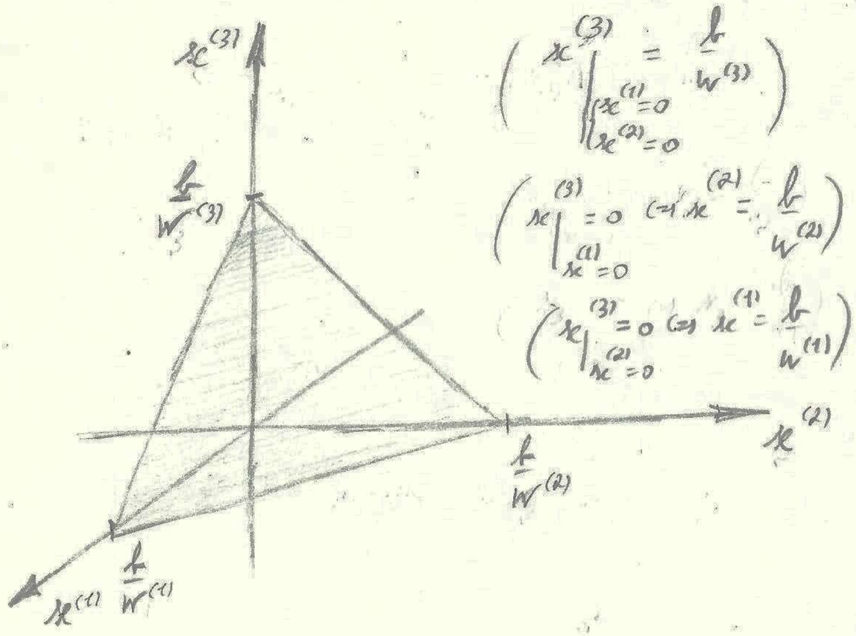
$D=2$:

$$w^{(1)}x^{(1)} + w^{(2)}x^{(2)} - b = 0 \Leftrightarrow x^{(2)} = -\frac{w^{(1)}}{w^{(2)}}x^{(1)} + \frac{b}{w^{(2)}}$$



$D=3$:

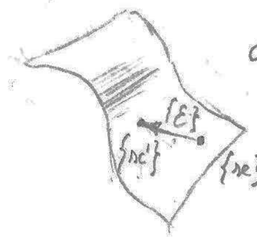
$$w^{(1)}x^{(1)} + w^{(2)}x^{(2)} + w^{(3)}x^{(3)} - b = 0 \Leftrightarrow x^{(3)} = -\frac{w^{(1)}}{w^{(3)}}x^{(1)} - \frac{w^{(2)}}{w^{(3)}}x^{(2)} + \frac{b}{w^{(3)}}$$



Let $\boxed{F = \{w\}^T\{x\} - b}$ (2). It is possible to show that the normal of the surface defined by $F=0$, has a constant normal. The normal of the surface is

$$\boxed{\{n\} = \frac{\{\nabla F\}}{\|\{\nabla F\}\|}} \quad (3)$$

Demonstration: ($\{n\} = \frac{\{\nabla F\}}{\|\{\nabla F\}\|}$ for a surface defined as $F=0$)



Since the surface is defined as $F(\{x\}) = 0 = \text{const}$ one has

$$F(\{x'\}) = F(\{x\}) \Leftrightarrow$$

$$\Leftrightarrow F(\{x\}+\{\varepsilon\}) - F(\{x\}) = 0 \Leftrightarrow$$ ← Taylor series expansion of the first term

$$\Leftrightarrow F(\{x\}) + \{\nabla F\}^T(\{x\})\cdot\{\varepsilon\} + \mathcal{O}(\|\{\varepsilon\}\|^2) - F(\{x\}) = 0 \Leftrightarrow$$

$$\lim_{\|\{\varepsilon\}\| \to 0} \{\nabla F\}^T(\{x\})\cdot\{\varepsilon\} = 0$$

Since in the limit $\|\{\varepsilon\}\| \to 0$, $\{\varepsilon\}$ is tangent to the surface, the condition above implies that $\{\nabla F\}$ is normal to $\{\varepsilon\}$ and so also to the surface.

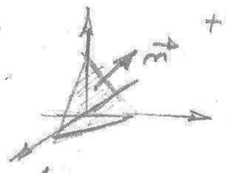
Therefore,

$$\|\{\nabla F\}\|\{n\} = \{\nabla F\} = \{\nabla\}\left(\{w\}^T\{x\} - b\right) \|\{\nabla F\}\| n^{(i)} = \frac{\partial}{\partial x^{(i)}}\Big/\left(\sum_j w^{(j)}x^{(j)} - b\right) \Leftrightarrow$$

$$\Leftrightarrow \|\{\nabla F\}\| n^{(i)} = \sum_j w^{(j)} \underbrace{\frac{\partial x^{(j)}}{\partial x^{(i)}}}_{=\delta_{ij}} \Leftrightarrow n^{(i)} = \frac{w^{(i)}}{\|\{w\}\|} \Leftrightarrow \boxed{\{n\} = \frac{\{w\}}{\|\{w\}\|} = \{\text{const}\}} \quad (4)$$

By having $\{n\} = \{\text{const}\}$ one infers that $\{w\}^T\{x\} - b$ has to define a hyperplane.

The SVM model is such that the label y predicted for feature $\{x\}$ is $+1$ if $\{x\}$ falls on the side of the domain divided by the decision boundary to which the conventional normal $\{m\}$ points to and -1 if $\{x\}$ falls on the other side. Mathematically, the prediction may be defined as



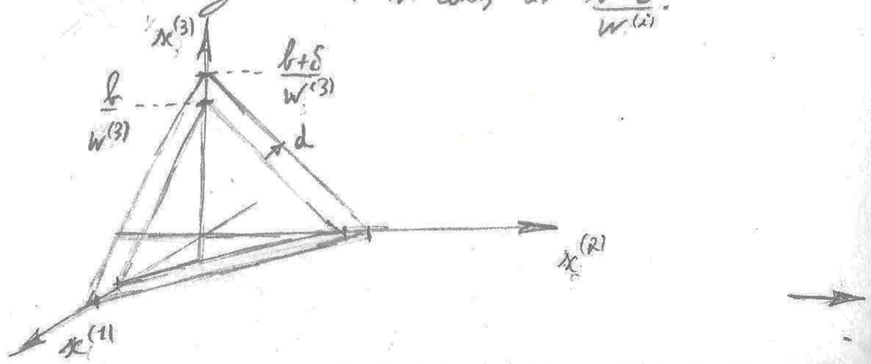
$$y = \begin{cases} -1, & \text{if } \{w\}^T\{x\} - b < 0 \\ +1, & \text{if } \{w\}^T\{x\} - b > 0 \end{cases} \quad \Leftrightarrow$$

$$\Leftrightarrow \boxed{y = f(\{x\}) = \text{sgn}(F(\{x\})) = \text{sgn}(\{w\}^T\{x\} - b)} \quad (5)$$

(SVM model)

Note:

One may geometrically understand what (5) represents by taking into account that $\{w\}^T\{x\} - b = 0$ corresponds to the decision boundary, a hyperplane of normal $\{m\} = \{\nabla F\}$ that intersects the i-th axis at $\frac{b}{w^{(i)}}$. The surface $\{w\}^T\{x\} - b = \delta \Leftrightarrow \{w\}^T\{x\} - (b+\delta) = 0$ corresponds to a hyperplane of same normal and intersecting the i-th axis at $\frac{b+\delta}{w^{(i)}}$.



One may associate to a feature $\{x_0\}$ a hyperplane of same normal as the decision boundary ($\{m\} = \{w\}$) that passes through this very point. Such hyperplane is defined by

$$\{w\}^T\{x\} - b_0 = 0$$

so that

$$\{w\}^T\{x_0\} - b_0 = 0 \Leftrightarrow b_0 = \{w\}^T\{x_0\}$$

It is straightforward to compute δ:

$$b + \delta := b_0 = \{w\}^T\{x_0\} \Leftrightarrow \delta := \{w\}^T\{x_0\} - b = F(\{x_0\}).$$

Therefore, the prediction for $\{x_0\}$ corresponds to $+1$, if its associated hyperplane has a bias b_0 greater than b and to -1, if its associated hyperplane has a bias smaller than b.

Note also that one may also compute the displacement of the hyperplane for $\{x_0\}$ from the decision boundary along the direction of $\vec{m}$ through the value of δ. Let such displacement correspond d. From the geometrical representation shown on the left-hand side, one infers

$$d = \left(\frac{b+\delta}{w^{(i)}} - \frac{b}{w^{(i)}}\right)\{e^{(i)}\}^T \underbrace{\{m\}}_{= \frac{\{w\}}{\|\{w\}\|} \text{ from (4)}} = \frac{\delta}{w^{(i)}}\{e^{(i)}\}^T \frac{\{w\}}{\|\{w\}\|} = \frac{\delta}{w^{(i)}} \frac{w^{(i)}}{\|\{w\}\|} = \frac{\delta}{\|\{w\}\|} \quad (\delta = F(\{x_0\}))$$

$$\Leftrightarrow \boxed{d = \frac{F(\{x_0\})}{\|\{w\}\|}} \quad (6)$$

(Displacement of hyperplane of $\{x_0\}$ along $\{m\}$ with respect to the decision boundary)

The most appropriate decision boundary would be one that makes all labelled examples to lay on their correct sides, if possible, and maximizes the distances between itself and the closest points from both sides. One may then consider two other hyperplanes defining a margin with the decision boundary exactly in the middle so that the labelled examples do not fall or dot margin, satisfying

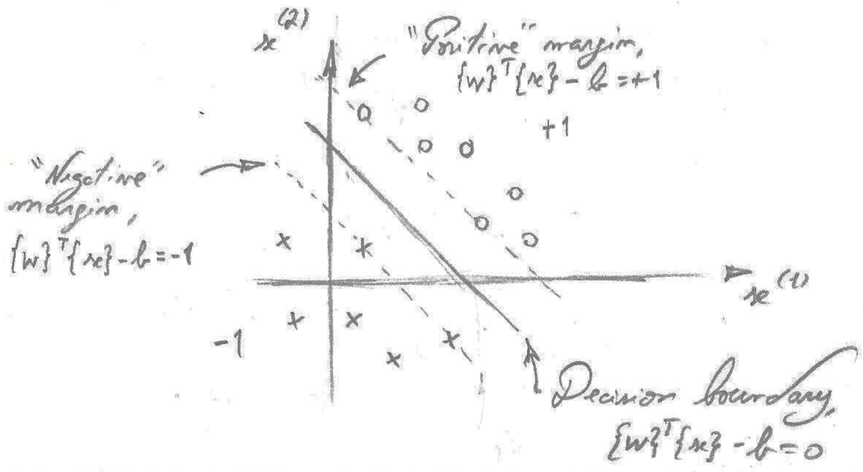
$$\begin{cases} \{w\}^T\{x_i\} - b \le c, & \text{if } y_i = -1 \\ \{w\}^T\{x_i\} - b \ge c, & \text{if } y_i = +1 \end{cases} \Leftrightarrow$$

$$\Leftrightarrow \begin{cases} \frac{\{w\}^T}{c}\{x_i\} - \frac{b}{c} \le 1, & \text{if } y_i = -1 \\ \frac{\{w\}^T}{c}\{x_i\} - \frac{b}{c} \ge 1, & \text{if } y_i = +1 \end{cases} \Leftrightarrow$$

(Change of notation: $\frac{\{w\}}{c} \leftrightarrow \{w\}$, $\frac{b}{c} \leftrightarrow b$)

$$\Leftrightarrow \boxed{y_i\left(\{w\}^T\{x_i\} - b\right) - 1 \ge 0} \quad (7)$$

("Hard-margin" constraint)



The best decision boundary would be one whose distance between a margin hyperplane and the boundary decision is the greatest one. Well, according to (6), this distance is given by

$$|d| = \frac{|F(\{x_0\})|}{\|\{w\}\|} \Leftrightarrow |d| = \frac{1}{\|\{w\}\|}.$$

($\{x_0\}$ is a point in the margin hyperplane. In this hyperplane $F(\{x_0\}) = \pm 1$)

In order to maximize $|d| = \frac{1}{\|\{w\}\|}$ one needs to minimize $\|\{w\}\|$, or equivalently $\frac{1}{2}\|\{w\}\|^2 = \frac{1}{2}\{w\}^T\{w\}$.

"Hard-margin" primal form of the SVM problem

Obtaining the SVM model corresponds to find the parameters $\{w\}$ and b that minimize $\frac{1}{2}\{w\}^T\{w\}$ subject to $y_i\left(\{w\}^T\{x_i\} - b\right) - 1 \ge 0$, that is,

$$\boxed{\begin{aligned} &\min_{\{w\},b} && \frac{1}{2}\{w\}^T\{w\} \\ &\text{subject to} \\ &&& y_i\left(\{w\}^T\{x_i\} - b\right) - 1 \ge 0 \end{aligned}} \quad (8)$$

("Hard-margin" primal form of the SVM problem)

Problem (8) is a Quadratic Linear Programming (QLP) type I problem. Also, it is said to be a "hard-margin" one

since that it assumes that all training points are out of the margin region (condition (7)), or, in other words, that the training set is linearly separable - there is a hyperplane that separates features of a class from features of the other. It is termed "primal", since there is actually another equivalent form (the "dual") for expressing the same problem, as discussed latter.

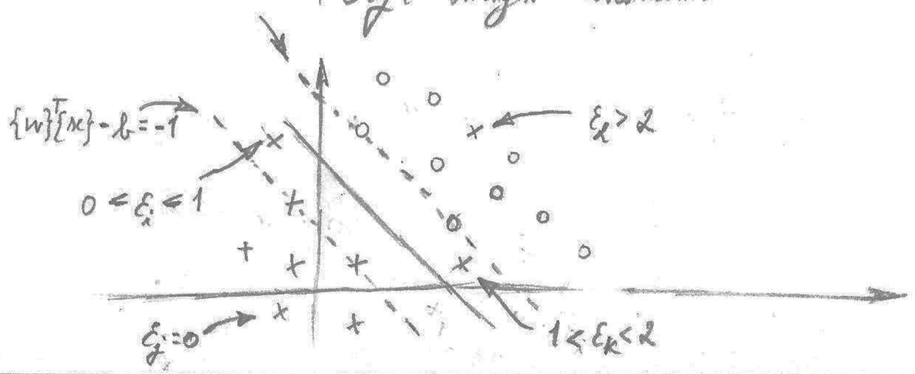
The term $\frac{1}{2}\{w\}^T\{w\}$ is called an "objective function" - a function whose absolute minimum or maximum is sought.

"Soft-margin" primal form of the SVM problem

One may relax condition (7) by considering new parameters $\xi_i \geq 0$, termed "slack variables", added to the left-hand side of (7), meaning that training points may be within the margin making it "soft", or even on the "incorrect" side of the decision boundary:

$$y_i(\{w\}^T\{x_i\} - b) - 1 + \xi_i \geq 0 \qquad (8)$$

$\{w\}^T\{x\} - b = +1$ ("soft-margin" constraint)



In this case, besides minimizing $\frac{1}{2}\{w\}^T\{w\}$, it would be also important to minimize ξ_i, so that one may get as close as possible to the "hard-margin" case.

An objective function for the "soft-margin" problem would naturally be

$$\frac{1}{2}\{w\}^T\{w\} + D\sum_{i=1}^{N}\xi_i$$

where D is a "hyperparameter" - pre-defined.

Note that:

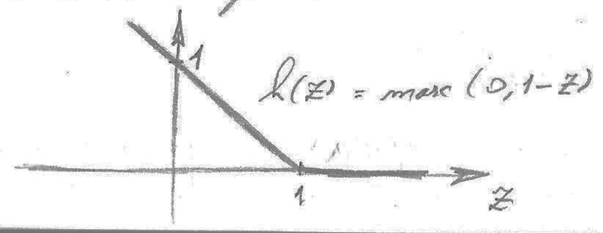
$$\xi_i = \begin{cases} 0, \text{ if } y_i(\{w\}^T\{x_i\} - b) - 1 \geq 0 & \text{(out of margin, and right side case)} \\ -y_i(\{w\}^T\{x_i\} - b) - 1), \text{ if } y_i(\{w\}^T\{x_i\} - b) - 1 < 0 & \text{(in-margin and/or wrong case)} \end{cases}$$

$$\Leftrightarrow \xi_i = \begin{cases} 0, \text{ if } 1 - y_i(\{w\}^T\{x_i\} - b) \leq 0 \\ 1 - y_i(\{w\}^T\{x_i\} - b), \text{ if } 1 - y_i(\{w\}^T\{x_i\} - b) > 0 \end{cases} \Leftrightarrow$$

$$\Leftrightarrow \boxed{\xi_i = \max\left(0,\ 1 - y_i(\{w\}^T\{x_i\} - b)\right)} \qquad (9)$$

("Slack variable")

One may write (9) as $\xi_i = \max(0, 1 - z) =: h(z)$ with $z = y_i(\{w\}^T\{x_i\} - b)$ which corresponds to the so-called "hinge loss function":



One may then write the "soft-margin" SVM model in two ways:

$$\min_{\{w\},b} \ \frac{1}{2}\{w\}^T\{w\} + \nu\,\frac{1}{N}\sum_{i=1}^{N} \max\left(0,\, 1 - y_i\left(\{w\}^T\{x_i\} - b\right)\right) \tag{10}$$

$$\min_{\{w\},b,\{\xi\}} \ \frac{1}{2}\{w\}^T\{w\} + \nu \sum_{i=1}^{N} \xi_i$$
$$\text{subject to} \begin{cases} y_i\left(\{w\}^T\{x_i\} - b\right) - 1 + \xi_i \geq 0 \\ \xi_i \geq 0 \end{cases} \tag{11}$$

("soft-margin" primal forms of the SVM problem)

Dual form of the SVM problem

Problem (11) may be rewritten in a "dual form", obtained through the method of Lagrange multipliers for the case of the constraints corresponding to inequalities - the so-called Karush-Kuhn-Tucker theorem (KKT).

The function to minimize corresponds to

$$\mathcal{L}(\{w\},\{\xi\}) = \frac{1}{2}\{w\}^T\{w\} + \nu\sum_{i=1}^{N}\xi_i \tag{12}$$

And the constraints correspond to

$$g_i(\{w\},b,\xi_i) = 1 - \xi_i - y_i\left(\{w\}^T\{x_i\} - b\right) \leq 0 \tag{13}$$

and

$$h_i(\xi_i) = -\xi_i \leq 0 \tag{14}$$

The Lagrangian of the problem is given by

$$\mathcal{L}(\{w\},b,\{\xi\},\{\alpha\},\{\beta\}) = f(\{w\},b,\{\xi\}) + \left(\sum_{i=1}^{N}\alpha_i g_i(\{w\},b,\xi_i)\right) + \left(\sum_{i=1}^{N}\beta_i h_i(\xi_i)\right) \tag{15}$$

where $\alpha_i \geq 0$ (16) and $\beta_i \geq 0$ (17) are the so-called Lagrange multipliers.

It is possible to show that (11) is equivalent to its dual form,

$$\min_{\{w\},b,\{\xi\}} \max_{\{\alpha\},\{\beta\}} \mathcal{L}(\{w\},b,\{\xi\},\{\alpha\},\{\beta\}) \Leftrightarrow$$

$$\Leftrightarrow \min_{\{w\},b,\{\xi\}} \max_{\{\alpha\},\{\beta\}} f(\{w\},b,\{\xi\}) + \left(\sum_{i=1}^{N}\alpha_i g_i(\{w\},b,\xi_i)\right) + \left(\sum_{i=1}^{N}\beta_i h_i(\xi_i)\right)$$

("soft-margin" dual form of the SVM problem) (16)

One may trivially know that for $g_i > 0$ and/or $h_i > 0$ (that is, for the case of the constraints not being satisfied) $\alpha_i = +\infty$ and/or $\beta_i = +\infty$ would maximize the Lagrangian $\mathcal{L}$. And for $g_i \leq 0$ and $h_i \leq 0$, $\alpha_i = 0$ and $\beta_i = 0$ would maximize $\mathcal{L}$, making this to correspond to f.

$$\min_{\{w\},b,\{\xi\}} \max_{\{\alpha\},\{\beta\}} \mathcal{L} = \min_{\{w\},b,\{\xi\}} \left(\begin{cases} +\infty, & \text{if } (\{w\},b,\{\xi\}) \in \mathbb{R}^{D+1+N} : g_i > 0 \vee h_i > 0 \\ f, & \text{if } (\{w\},b,\{\xi\}) \in \mathbb{R}^{D+1+N} : g_i \leq 0 \wedge h_i \leq 0 \end{cases} \right) \Leftrightarrow$$

$$\Leftrightarrow \min_{\{w\},b,\{\xi\}} f \quad \text{subject to } g_i \leq 0 \wedge h_i \leq 0 \ \forall i = 1,\ldots,N$$

One has then shown that (16) is equivalent to (11). Moreover, the optimum parameters $\{w^*\}$, b^*, $\{\xi^*\}$, α^* and β^* would be such that:

$$\mathcal{L}(\{w^*\}, b^*, \{\xi^*\}) + \left(\sum_{i=1}^{N} \alpha_i^* g_i(\{w^*\}, b^*, \xi_i^*)\right) + \left(\sum_{i=1}^{N} \beta_i^* h_i(\xi_i^*)\right) = f(\{w^*\}, b^*, \{\xi^*\}) \Leftrightarrow$$

$$\Leftrightarrow \boxed{\sum_{i=1}^{N} \alpha_i^* g_i(\{w^*\}, b^*, \xi_i^*) + \sum_{i=1}^{N} \beta_i^* h_i(\xi_i^*) = 0} \quad (17)$$

Since $\alpha_i \geq 0$, $\beta_i \geq 0$ and $g_i \leq 0$, $h_i \leq 0$, each term in the series is non-positive. But for (17) to hold, each term actually needs to be null, that is

$$\begin{cases} \boxed{\alpha_i^* g_i(\{w^*\}, b^*, \xi_i^*) = 0 \quad \forall i = 1, \dots, N} & (18) \\ \boxed{\beta_i^* h_i(\xi_i^*) = 0 \quad \forall i = 1, \dots, N} & (19) \end{cases}$$

(Complementary slackness conditions)

Now, to satisfy (18) and (19) one has to have $\alpha_i^* = 0$ if $g_i < 0$, $g_i = 0$ if $\alpha_i^* > 0$; $\beta_i^* = 0$ if $h_i < 0$ and $h_i = 0$ if $\beta_i^* > 0$.

Problem (11) is a convex one (the minimizing and the constraint functions are convex – their second order derivatives are non-negative), and for that case the min and max operators in (16) may be swapped.

Therefore, (16) may be transformed into

$$\boxed{\max_{\{\alpha\},\{\beta\}} \min_{\{w\}, b, \{\xi\}} \mathcal{L}(\{w\}, b, \{\xi\}) + \left(\sum_{i=1}^{N} \alpha_i g_i(\{w\}, b, \xi_i)\right) + \left(\sum_{i=1}^{N} \beta_i h_i(\xi_i)\right)} \quad (20)$$

$$= \mathcal{L}(\{w\}, b, \{\xi\}, \{\alpha\}, \{\beta\})$$

Let's minimize $\mathcal{L}$ with respect to $\{w\}, b, \{\xi\}$. Well,

$$\boxed{\mathcal{L} = \frac{1}{2}\{w\}^T\{w\} + D\sum_{i=1}^{N} \xi_i + \left(\sum_{i=1}^{N} \alpha_i \left(1 - \xi_i - y_i(\{w\}^T\{x_i\} - b)\right)\right) - \left(\sum_{i=1}^{N} \beta_i \xi_i\right)} \quad (21)$$

$$\frac{\partial \mathcal{L}}{\partial w^{(j)}} = 0 \Leftrightarrow \frac{1}{2} \times 2 w^{(j)} - \sum_{i=1}^{N} \alpha_i y_i \times x_i^{(j)} = 0 \Leftrightarrow w^{(j)} - \left(\sum_{i=1}^{N} \alpha_i y_i x_i^{(j)}\right) = 0 \Leftrightarrow$$

$$\left(\{w\}^T\{w\} = \sum_{k=1}^{D} w^{(k)2} \Rightarrow \frac{\partial(\{w\}^T\{w\})}{\partial w^{(j)}} = \sum_{k=1}^{D} \frac{\partial\left(w^{(k)2}\right)}{\partial w^{(j)}} = 2w^{(j)}\right)$$

$$\left(\{w\}^T\{x_i\} = \sum_{k=1}^{D} w^{(k)} x_i^{(k)} \quad \frac{\partial(\{w\}^T\{x_i\})}{\partial w^{(j)}} = \sum_{k=1}^{D} \frac{\partial w^{(k)}}{\partial w^{(j)}} x_i^{(k)} = x_i^{(j)}\right)$$

$$\boxed{\{w\} = \sum_{i=1}^{N} \alpha_i y_i \{x_i\}} \quad (22)$$

("Soft-margin" optimum weight vector)

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Leftrightarrow \boxed{\sum_{i=1}^{N} \alpha_i y_i = 0} \quad (23)$$

$$\frac{\partial \mathcal{L}}{\partial \xi_j} = 0 \Leftrightarrow \mathcal{D} - \alpha_j - \beta_j = 0 \Leftrightarrow \boxed{\beta_j = \mathcal{D} - \alpha_j} \tag{24}$$

One may now rephrase (20) as

$$\max_{\{\alpha\},\{\beta\}} \mathcal{L}\Big|_{\substack{(22)\\(23)\\(24)}} = \text{subject to (23) and (24)} =$$

$$= \max_{\{\alpha\},\{\beta\}} \left(\frac{1}{2} \left(\sum_{i=1}^{N} \alpha_i y_i \{x_i\}^T \right) \left(\sum_{j=1}^{N} \alpha_j y_j \{x_j\} \right) + \mathcal{D} \sum_{i=1}^{N} \xi_i + \right.$$

$$\left. + \sum_{i=1}^{N} \alpha_i \left(1 - \xi_i - y_i \left(\left(\sum_{j=1}^{N} \alpha_j y_j \{x_j\}^T \right) \{x_i\} - b \right) \right) - \sum_{i=1}^{N} (\mathcal{D} - \alpha_i) \xi_i \right)$$

$$\text{subject to } \sum_{i=1}^{N} \alpha_i y_i = 0 \text{ and } \beta_j = \mathcal{D} - \alpha_j$$

$$= \max_{\{\alpha\},\{\beta\}} \left(\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \{x_i\}^T \{x_j\} + \cancel{\mathcal{D} \sum_{i=1}^{N} \xi_i} + \right.$$

$$+ \sum_{i=1}^{N} \alpha_i - \cancel{\sum_{i=1}^{N} \alpha_i \xi_i} - \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \{x_j\}^T \{x_i\}$$

$$\left. + b \underbrace{\sum_{i=1}^{N} \alpha_i y_i}_{=0} - \cancel{\mathcal{D} \sum_{i=1}^{N} \xi_i} + \cancel{\sum_{i=1}^{N} \alpha_i \xi_i} \right)$$

$$\text{subject to } \sum_{i=1}^{N} \alpha_i y_i = 0 \text{ and } \beta_j = \mathcal{D} - \alpha_j$$

$$= \max_{\{\alpha\},\{\beta\}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \{x_i\}^T \{x_j\}$$

$$\text{subject to } \sum_{i=1}^{N} \alpha_i y_i = 0 \text{ and } \beta_j = \mathcal{D} - \alpha_j$$

(There is no β_j in the function to be maximized. Therefore, β_j may take any non-negative value. From (24), $\alpha \geq 0$ and $\beta \geq 0$ conditions, one gets $0 \leq \alpha_j \leq \mathcal{D}$.)

$$= \boxed{\begin{aligned} &\max_{\{\alpha\}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \{x_i\}^T \{x_j\} \\ &\text{subject to} \\ &\qquad 0 \leq \alpha_i \leq \mathcal{D} \qquad \forall i = 1, \dots, N \\ &\qquad \sum_{i=1}^{N} \alpha_i y_i = 0 \end{aligned}} \tag{25}$$

("Soft-margin" dual form of the SVM problem).

Let one understand the influence of the values of α_i on the solution of (25).

- Case 1: $\alpha_i = 0$
  - From (24) one gets $\beta_i = \mathcal{D}$.
  - From (19) one gets $h_i(\xi_i) = 0 \Leftrightarrow \xi_i = 0$.
  - From (13) one gets $y_i (\{w\}^T \{x_i\} - b) \geq 1$.

  Therefore for $\alpha_i = 0$, the label for the feature $\{x_i\}$ is correctly predicted and the feature is either on the correct side of the respective margin hyperplane or precisely on this hyperplane.

Case 2: $0 < \alpha_i < \nu$

From (24), one gets $0 < \beta_i < \nu$

From (19), one gets $\xi_i = 0$

From (18) one gets $g_i(\{w\}, b, \xi_i) = 0 \Leftrightarrow$

$\Leftrightarrow y_i(\{w\}^T\{x_i\} - b) = 1$

Therefore, for $0 < \alpha_i < \nu$, the label for $\{x_i\}$ is correctly predicted and the feature is precisely on the respective margin hyperplane.

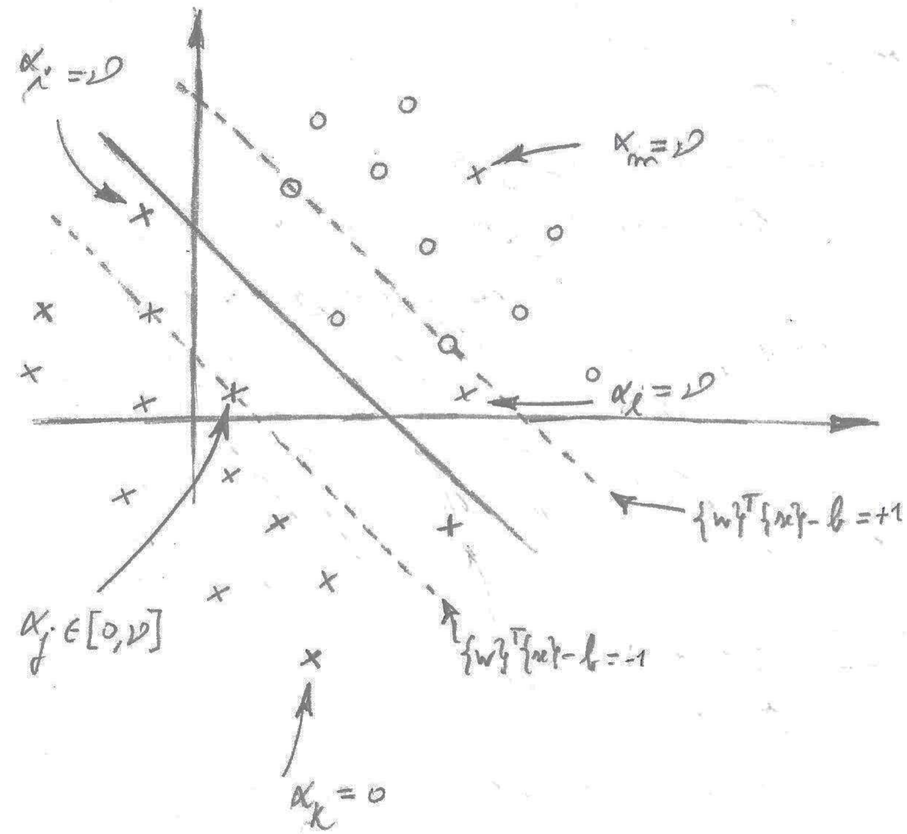
Case 3: $\alpha_i = \nu$

$\xi_i \geq 0$

From (18) one gets $g_i(\{w\}, b, \xi_i) = 0 \Leftrightarrow$

$\Leftrightarrow y_i(\{w\}^T\{x_i\} - b) = 1 - \xi_i$

Therefore, for $\alpha_i = \nu$, the label for $\{x_i\}$ might or not be correctly predicted. Depending on the value of ξ_i the feature is either precisely on the respective margin hyperplane, in-margin and correct side, in-margin and incorrect side or out of margin at the incorrect side of the decision boundary.



One may compute the optimum b value from a data point for which $\alpha_i > 0$. In this case the complementary slackness condition (18) implies that

$$g_i(\{w\}, b, \xi_i) = 0 \wedge \alpha_i > 0 \overset{(13)}{\Leftrightarrow} 1 - \xi_i - y_i(\{w\}^T\{x_i\} - b) = 0$$

$$\wedge \alpha_i > 0 \overset{(22)}{\Leftrightarrow} 1 - \xi_i - y_i\left(\sum_{j=1}^{N} \alpha_j y_j \{x_j\}^T\{x_i\} - b\right) = 0 \wedge \alpha_i > 0 \Leftrightarrow$$

$$\Leftrightarrow b = \left(\sum_{j=1}^{N} \alpha_j y_j \{x_j\}^T\{x_i\}\right) + \frac{1 - \xi_i}{y_i}, \wedge \alpha_i = 0 \Leftrightarrow$$

$$\left(\text{since } y_i \in \{-1, 1\} \overset{1 =}{\Rightarrow} \frac{1}{y_i} = y_i\right)$$

$$\boxed{b = \left(\sum_{j=1}^{N} \alpha_j y_j \{x_j\}^T\{x_i\}\right) - y_i(1 - \xi_i), \text{ for an } \alpha_i > 0} \quad (26)$$

("soft-margin" optimum bias)

Support vectors

One may understand from (22) and (26) that solely data points for which α_i is non-null contribute for the calculation of the optimum parameters $\{w\}$ and b. This points are then called **support vectors**, since they are the ones that support for the calculation of such parameters.

From the figure in the previous page one may conclude that the support vectors corresponding to points precisely on the margin hyperplanes for which $\alpha_i > 0$, and points inside the margin, and points out of the margin on the incorrect side.

"Hard-margin" case

The "hard-margin" case may be obtained from the "soft-margin" one by giving an infinitely large weight to the term of the objective function,

$$\frac{1}{2}\{w\}^T\{w\} + \nu \sum_{i=1}^{N} \xi_i ,$$

associated with the slack variables ξ_i, that is, $\nu \sum_{i=1}^{N} \xi_i$. That may be done by choosing the hyperparameter ν to be $\nu = +\infty$. In such conditions, the minimization of the objective function would result in $\xi_i = 0 \ \forall \ i = 1, \dots, N$, making the margin "hard".

The problem (25) is transformed into one for which $\nu = +\infty$, that is,

$$\max_{\{\alpha\}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \{x_i\}^T \{x_j\}$$

subject to

$$0 \leq \alpha_i \leq +\infty, \quad \forall \ i = 1, \dots, N$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

("Hard-margin" dual form of the SVM problem) (27)

The expression for the optimum $\{w\}$ parameter remains the same;

$$\{w\} = \sum_{i=1}^{N} \alpha_i y_i \{x_i\}$$

(28)

("Hard-margin" optimum weight vector)
(coinciding with (22))

The expression for optimum b parameter, (26), is transformed into one for which $\xi_i = 0 \ \forall \ i = 1 \dots N$, that is,

$$b = \left(\sum_{j=1}^{N} \alpha_j y_j \{x_j\}^T \{x_i\} \right) - y_i, \text{ for an } \alpha_i > 0$$

(29)

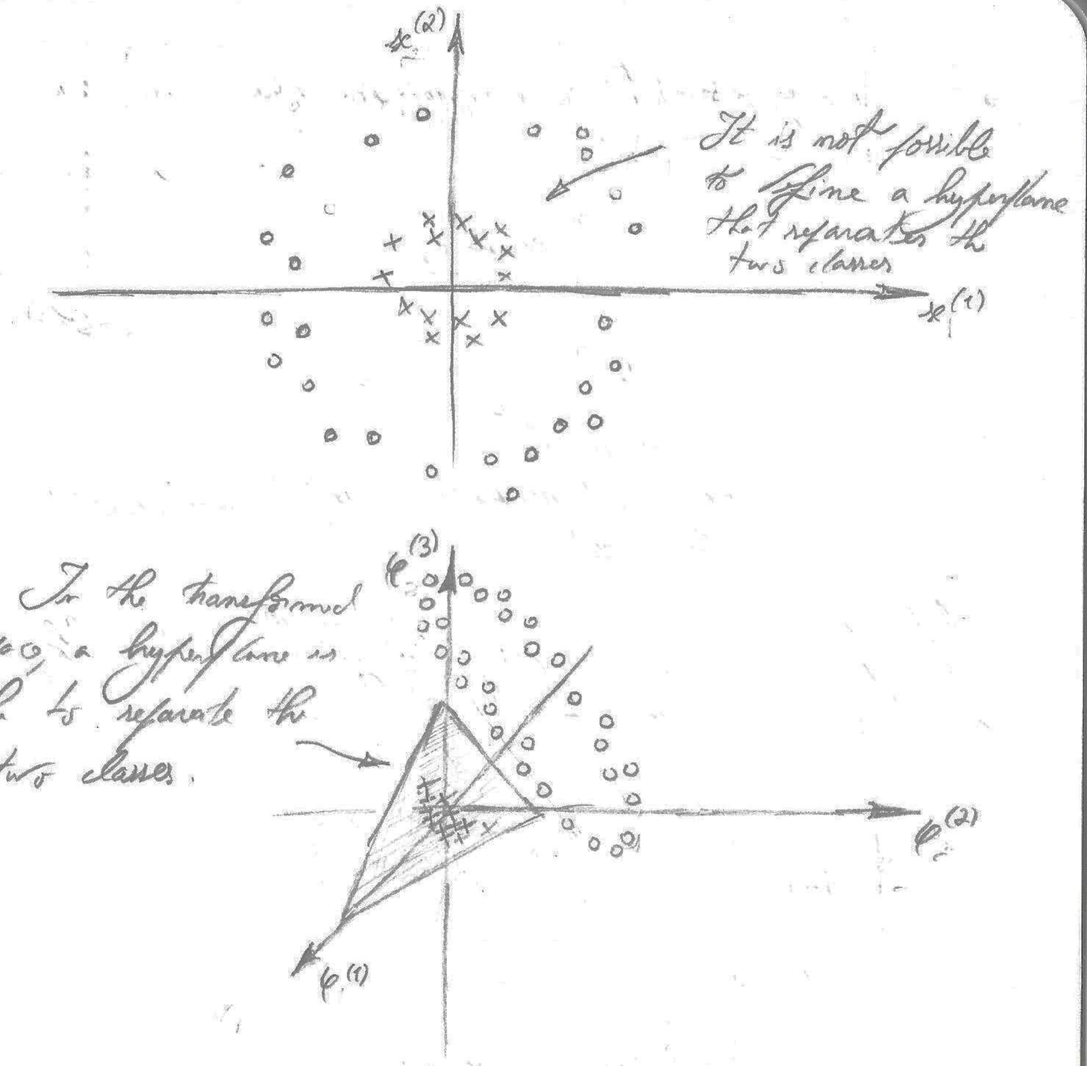
("Hard-margin" optimum bias)

Note since for the "hard-margin" case all data points lay on the correct side of the decision boundary and out of the margins, the supporting vectors are now data points that are precisely on the margin hyperplanes and $\alpha_i > 0$.

It is important to understand that the "hard-margin" case only holds if the data points are linearly separable, that is, there is a hyperplane that separates all data points associated with one label from all data points associated with the other.

Kernelisation

For the case of the data points not being linearly separable, there is the possibility of transforming the space into a new one by applying a mapping $\{\varphi\}: \{x\} \to \{\varphi(\{x\})\}$ so that in the latter the data points are linearly separable and a hyperplane may effectively separate the two classes — this is the so-called kernel trick. Consider as example the case depicted in the figure of the following page: the data points are not linearly separable in the original space but they are in the transformed one.



The usage of the mapping φ may be expressed by substituting every $\{x\}$ instance in SVM relations by $\{\varphi\}$. The SVM model (5) is transformed into

$$y = \operatorname{sgn}\left(\{w\}^T\{\varphi\} - b\right) = \operatorname{sgn}\left(\left(\sum_{j=1}^{N} \alpha_j y_j \{\varphi_j\}^T\{\varphi\}\right) - \left(\sum_{j=1}^{N} \alpha_j y_j \{\varphi_j\}^T\{\varphi_i\} - y_i\right)\right)$$

("soft-margin" relation for optimum $\{w\}$, (22))

("soft-margin" relation for optimum b, (26))

for an $\alpha_i > 0$ (=)

(2)

$$y = \operatorname{sgn}\left(\left(\sum_{j=1}^{N} \alpha_j y_j \left(\{\varphi_j\}^T\{\varphi\} - \{\varphi_j\}^T\{\varphi_i\}\right)\right) + y_i\right)$$

for any $\alpha_i > 0$

(30)

(SVM "soft-margin" model in the case of the feature vector $\{x\}$ being subject to a map $\{\varphi\}:\{x\} \to \{\varphi(\{x\})\}$)

And the "soft-margin" dual form of the SVM problem is transformed into

$$\max_{\{\alpha\}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \{\varphi_i\}^T\{\varphi_j\}$$

subject to

$$0 \le \alpha_i \le \frac{1}{2N} \qquad \forall\, i = 1, \dots, N$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

(31)

("soft-margin" dual form of the SVM problem in the case of the feature vector $\{x\}$ being subject to a map $\{\varphi\}:\{x\} \to \{\varphi(\{x\})\}$)

From (30) and (31) one may conclude that both SVM "soft-margin" model and problem in the case of the feature vector $\{x\}$ being subject to a map $\{\varphi\}$ would depend on the latter through the products $\{\varphi_i\}^T\{\varphi\}$ and $\{\varphi_i\}^T\{\varphi_j\}$. Therefore there is no need for one to compute the individual transformations $\{x_i\} \to \{\varphi(\{x_i\})\}$ but the product transformations $\{x_i\}^T\{x_j\} \to \{\varphi(\{x_i\})\}^T\{\varphi(\{x_j\})\}$. A function defined as

$$K : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$$
$$(\{x_i\}, \{x_j\}) \to K(\{x_i\}, \{x_j\}) = \{\varphi(\{x_i\})\}^T\{\varphi(\{x_j\})\},$$

(Positive-definite Kernel function) (32)

corresponding to a positive-definite kernel function, would produce such transformation. A function is a positive-definite kernel function, if it satisfy the conditions of Mercer's theorem.

(Mercer's theorem)

Let $K(\{x_i\}, \{x_j\})$ be symmetric (that is, $K(\{x_i\}, \{x_j\}) = K(\{x_j\}, \{x_i\})$) and continuous. $K(\{x_i\}, \{x_j\})$ would be a positive-definite Kernel function, that is, satisfying (32), if the respective Gram matrice, $[K]$ (defined so that $K^{(i,j)} = K(\{x_i\}, \{x_j\})$) is positive semi-definite.

Note: (Symmetric function)

The kernel function needs to be symmetric since the inner product between two vectors is also symmetric:

$$\{\varphi_i\}^T\{\varphi_j\} = \sum_k \varphi_i^{(k)}\varphi_j^{(k)} = \sum_k \varphi_j^{(k)}\varphi_i^{(k)} = \{\varphi_j\}^T\{\varphi_i\}$$

Note: (Positive semi-definite matrix)

A real symmetric matrix $[A]$ is positive semi-definite, if and only if

$$\{x\}^T[A]\{x\} \geq 0 \quad \forall \{x\} \in \mathbb{R}^D \setminus \{0\}$$

or equivalently, all of its eigenvalues λ_k are non-negative.

One may show the above statement through reversible implication:

- $[A]$ being positive semi-definite $\Rightarrow \lambda_k \geq 0 \ \forall K=1,\dots,D$

  If $\{v_k\}$ corresponds to a eigenvector of $[A]$, that is, $\{v_k\}$ then,

  $$\{v_k\}^T[A]\{v_k\} = \{v_k\}^T\lambda_k\{v_k\} = \lambda_k \underbrace{\|\{v_k\}\|^2}_{\geq 0} \geq 0,$$

  since $[A]$ is semi-definite positive. And because $\|\{v_k\}\|^2 \geq 0$ the above condition implies that $\lambda_k \geq 0$.

- $\lambda_k \geq 0 \ \forall K=1,\dots,D \Rightarrow$ $[A]$ being positive semi-definite

  A real symmetric matrix has an orthogonal matrix as a diagonalizing matrix, that is, $[S] = [Q]$.

Therefore,

$$[A] = [S][\Lambda][S]^{-1} = [Q][\Lambda][Q]^{-1} = [Q][\Lambda][Q]^T$$

($[\Lambda]$: diagonal matrix; $[Q]^{-1} = [Q]^T$: an orthogonal matrix is such that)

And,

$$\{x\}^T[A]\{x\} = \{x\}^T[Q][\Lambda][Q]^T\{x\} = \underbrace{([Q]^T\{x\})^T}_{=:\{z\}^T}[\Lambda]\underbrace{([Q]^T\{x\})}_{=:\{z\}} =$$

$$= \{z\}^T[\Lambda]\{z\} = \{z\}^T\Big(\sum_{ij}\underbrace{\lambda^{(i)}}_{=\lambda_i\delta_{ij}} z^{(j)}\{e_i\}\Big) =$$

$$= \{z\}^T\sum_i \lambda_i z^{(i)}\{e_i\} = \sum_i z^{(i)}\lambda_i z^{(i)} = \sum_i \underbrace{\lambda_i}_{\geq 0}\underbrace{\left(z^{(i)}\right)^2}_{\geq 0} \geq 0$$

As a conclusion,

$[A]$ being positive semi-definite

$\Leftrightarrow$

$\lambda_k \geq 0 \ \forall K=1,\dots,D.$

Note: (A real symmetric matrix has an orthogonal matrix as diagonalizing matrix)

One may prove that a real symmetric matrix has an orthogonal matrix as diagonalizing matrix through induction.

$n=1$

A real symmetric matrix 1×1, is already diagonal. Let this matrix be $[A] = [a]$ where a is a constant. The unity $[1]$ is a diagonalizing matrix of $[A]$:

$$[1][A][1]^{-1} = [1][a][1] = a$$

The unity is an orthogonal matrix since $[1]^{-1} = [1]^T = 1$.

Let one now show that a $n \times n$ real symmetric matrix is orthogonally diagonalizable if a $(n-1)\times(n-1)$ one also is.

Let $[A]$ be a $n \times n$ real symmetric matrix, with $n > 1$. Also, let λ_1 and $\{v_1\}$ be an eigenvalue and an eigenvector found for $[A]$. From $\{v_1\}$, one may an orthonormal basis for $\mathbb{R}^n$: $\mathcal{B} = \{\{v_1\} \dots \{v_n\}\}$.

Let $[P_{\mathcal{B}}] = [\{v_1\}\ \{v_2\} \cdots \{v_n\}]$ be a matrix with the elements of $\mathcal{B}$ as columns. Because $[P]$ is orthogonal, $[P]^{-1} = [P]^T$. This may be shown though the reasoning

$$[P]^T[P] = \begin{bmatrix} -\{v_1\}^T- \\ \vdots \\ -\{v_n\}^T- \end{bmatrix} \begin{bmatrix} | & & | \\ \{v_1\} & \cdots & \{v_n\} \\ | & & | \end{bmatrix} = \begin{bmatrix} \{v_1\}^T\{v_1\} & \cdots & \{v_1\}^T\{v_n\} \\ \vdots & \ddots & \\ \{v_n\}^T\{v_1\} & & \{v_n\}^T\{v_n\} \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \\ 0 & & 1 \end{bmatrix}$$

$$= [I] \quad (\Rightarrow$$

$$\Rightarrow [P]^T = [P]^{-1}.$$

Now, let one have a look on the matrix $[P]^{-1}[A][P]$.

- $[P]^{-1}[A][P]$ is symmetric because $([\underbrace{P}_{=[P]^T}]^{-1}[A][P])^T = [P]^T\underbrace{[A]^T}_{=[A]}[\underbrace{P}_{=[P]^{-1}}]^{TT} = [P]^{-1}[A][P]$.

- Its first column is $([P]^{-1}[A][P])\{e_1\} = [P]^{-1}[A]\{v_1\} = [P]^T\lambda_1\{v_1\} = \lambda_1 \begin{bmatrix} -\{v_1\}^T- \\ \vdots \\ -\{v_n\}^T- \end{bmatrix} \{v_1\} = \lambda_1 \begin{Bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{Bmatrix} = \begin{Bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{Bmatrix}$.

And since it was already showed that $[P]^{-1}[A][P]$ is symmetric, one may write this matrix as

$$[P]^{-1}[A][P] = \begin{bmatrix} \lambda_1 & 0 \cdots & 0 \\ 0 & & \\ \vdots & & [B] \\ 0 & & \end{bmatrix},$$

where $[B]$ needs to be a real symmetric $(n-1)\times(n-1)$ matrix. According to the premise, matrix $[B]$ is orthogonally diagonalizable and may be written as $[B] = [Q][\Lambda][Q]^{-1}$, when $[Q]$ is an orthogonal matrix and $[\Lambda]$ is a diagonal one.

Let

$$[R] = \begin{bmatrix} 1 & 0 \cdots & 0 \\ 0 & & \\ \vdots & & [Q] \\ 0 & & \end{bmatrix}.$$

One has

$$\begin{bmatrix} 1 & 0 \cdots & 0 \\ 0 & & \\ \vdots & & [Q] \\ 0 & & \end{bmatrix} \begin{bmatrix} 1 & 0 \cdots & 0 \\ 0 & & \\ \vdots & & [Q]^{-1} \\ 0 & & \end{bmatrix} = \begin{bmatrix} 1 & 0 \cdots & 0 \\ 0 & & \\ \vdots & & [Q][Q]^{-1} \\ 0 & & \end{bmatrix} = \begin{bmatrix} 1 & 0 \cdots & 0 \\ 0 & & \\ \vdots & & [I] \\ 0 & & \end{bmatrix} = [I].$$

So,

$$[R]^{-1} = \begin{bmatrix} 1 & 0 \cdots & 0 \\ 0 & & \\ \vdots & & [Q]^{-1} \\ 0 & & \end{bmatrix}.$$

Since $[Q]$ is an orthogonal matrix, $[Q]^T = [Q]^{-1}$, and $[R]^T = [R]^{-1}$. Therefore $[R]$ is also orthogonal.

Let

$$[U] = [P][R].$$

←

Matrix $[U]$ is orthogonal since the product of two orthogonal matrices is orthogonal:

$$([P][R])^T([P][R]) = [R]^T\underbrace{[P]^T[P]}_{=[I]}[R] = [R]^T[R] = [I].$$

Well,

$$[U]^{-1}[A][U] = ([P][R])^{-1}[A]([P][R]) = \leftarrow$$

$$= [R]^{-1}[P]^{-1}[A][P][R] =$$

$$= \begin{bmatrix} 1 & 0 \cdots 0 \\ 0 & \\ \vdots & [Q]^{-1} \\ 0 & \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \cdots 0 \\ 0 & \\ \vdots & [B] \\ 0 & \end{bmatrix} \begin{bmatrix} 1 & 0 \cdots 0 \\ 0 & \\ \vdots & [Q] \\ 0 & \end{bmatrix} =$$

$$= \begin{bmatrix} \lambda_1 & 0 \cdots 0 \\ 0 & \\ \vdots & [Q]^{-1}[B] \\ 0 & \end{bmatrix} \begin{bmatrix} 1 & 0 \cdots 0 \\ 0 & \\ \vdots & [Q] \\ 0 & \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \cdots 0 \\ 0 & \\ \vdots & [Q]^{-1}[B][Q] \\ 0 & \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \cdots 0 \\ 0 & \\ \vdots & [\Lambda'] \\ 0 & \end{bmatrix}.$$

$$([Q]^{-1}[B][Q] = [\Lambda'])$$

$$\left(\begin{array}{l} [R]^{-1}[P]^{-1}[P][R] = \\ = [R]^{-1}[R] = [I]. \\ \text{Therefore, } ([P][R])^{-1} = \\ [R]^{-1}[P]^{-1} \end{array} \right)$$

So, $[A] = [U]\begin{bmatrix} \lambda_1 & 0 \cdots 0 \\ 0 & \\ \vdots & [\Lambda'] \\ 0 & \end{bmatrix}[U]^{-1}$. Since $[D] = \begin{bmatrix} \lambda_1 & 0 \cdots 0 \\ 0 & \\ \vdots & [\Lambda'] \\ 0 & \end{bmatrix}$ is a diagonal matrix and $[U]$ is a orthogonal matrix, one may conclude that $[A]$ is orthogonally diagonalizable with

$$[A] = [U][D][U]^T$$

The homogeneous polynomial kernel

Let $\{x\}, \{z\} \in \mathbb{R}^D$. The homogeneous polynomial kernel is defined as

$$\boxed{K(\{x\},\{z\}) = (\{x\}^T\{z\})^d, \text{ with } d \in \mathbb{N}^+} \quad (33)$$

(Homogeneous polynomial kernel)

Note that

$$K(\{x\},\{z\}) = (\{x\}^T\{z\})^d = (x^{(1)}z^{(1)} + \cdots + x^{(D)}z^{(D)})^d =$$

$$= \sum_{\substack{m_1+\cdots+m_D=d \\ m_1,\ldots,m_D \geq 0}} \binom{d}{m_1,\ldots,m_D} (x^{(1)}z^{(1)})^{m_1} \cdots (x^{(D)}z^{(D)})^{m_D}$$

(Multinomial theorem)

$$(\Rightarrow K(\{x\},\{z\}) = \sum_{\substack{m_1+\cdots+m_D=d \\ m_1,\ldots,m_D \geq 0}} \left(\underbrace{\sqrt{\binom{d}{m_1,\ldots,m_D}} (x^{(1)})^{m_1} \cdots (x^{(D)})^{m_D}}_{= \varphi_j(\{x\})} \right) \times$$

$$\times \left(\underbrace{\sqrt{\binom{d}{m_1,\ldots,m_D}} (z^{(1)})^{m_1} \cdots (z^{(D)})^{m_D}}_{= \varphi_j(\{y\})} \right) =$$

$$= \{\varphi(\{x\})\}^T\{\varphi(\{y\})\},$$

where $\binom{d}{m_1,\ldots m_D} = \frac{d!}{m_1!\cdots m_D!}$ is the multinomial coefficient.

The sum is done over j, the index associated with a set of exponents $\{m_i\}_{i=1}^{D} = \{m_1, \ldots, m_D\}^D$ that satisfy $\sum_{i=1}^{D} m_i = m_1 + \ldots + m_D = d$.

The j-th component of the mapping function applied to the feature vector $\{x\}$ corresponds to

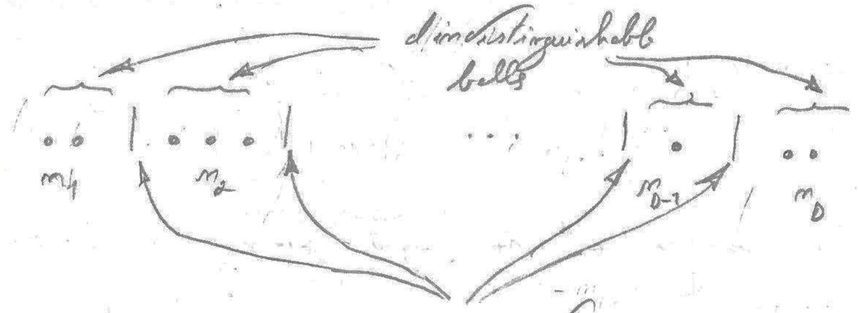
$$\varphi_j(\{x\}) = \sqrt{\binom{d}{m_1, \ldots, m_D}} (x^{(1)})^{m_1} \ldots (x^{(D)})^{m_D} \,. \tag{34}$$

The vector $\{\varphi(\{x\})\}$ may be expressed as

$$\{\varphi(\{x\})\} = \left\{ \sqrt{\binom{d}{m_1, \ldots, m_D}} (x^{(1)})^{m_1} \ldots (x^{(D)})^{m_D} \right\}^{m_1 + \ldots + m_D = d}_{m_1, \ldots, m_D \geq 0} \tag{35}$$

(Mapping function associated with the homogeneous polynomial kernel)

It would be interesting to determine the dimensionality of the mapping function $\{\varphi\}$. Well, this dimensionality corresponds to the number of different sets $\{m_i\}_{i=1}^{D}$ that satisfy $\sum_{i=1}^{D} m_i = d$. It is equivalent to number of ways of distributing d indistinguishable balls into D distinguishable boxes.



D-1 indistinguishable separators

There would be D-1 boundaries separating the D boxes from each other. The number of ways of distributing the d balls into the D boxes would be the same as the number of ways of arranging all items (balls and separators) in a line, $(d + D - 1)!$, if the balls were distinguishable and the boxes could have a different order. However, since the balls are indistinguishable and the order of the boxes is fixed, one needs to divide that number by the number of ways of ordering the balls and the separators between themselves, that is, $d!$ and $(D-1)!$, respectively.

The dimensionality of the mapping function is then

$$\frac{(d+D-1)!}{d!(D-1)!} = \binom{d+D-1}{d} = C_d^{d+D-1} \,, \tag{36}$$

(Dimensionality of the mapping function associated with the homogeneous polynomial kernel)

where $\binom{d+D-1}{d}$ is the binomial coefficient and C_d^{d+D-1} are the number of combinations without repetition of d elements which are possible to construct from a set of $d+D-1$ elements.

Note: (Multinomial theorem)

Theorem:

For any positive integer m and non-negative integer n, the n-th power of the sum of m monomials $x^{(1)}, \dots, x^{(m)}$ corresponds to

$$\left(x^{(1)} + \dots + x^{(m)}\right)^n = \sum_{\substack{k_1 + \dots + k_m = n \\ k_1, \dots, k_m \geq 0}} \binom{n}{k_1, \dots, k_m} \prod_{t=1}^{m} \left(x^{(t)}\right)^{k_t}, \tag{37}$$

where $\binom{n}{k_1, \dots, k_m} = n! / (k_1! \dots k_m!)$ (38) is the multinomial coefficient.

Note that the term "multinomial" is an agglutination of the Latin terms "multi-" (meaning "many much") and "nomius" (meaning "name"). It describes the terms that are subject to a power.

Demonstration:

The multinomial theorem may be proved through induction and the binomial theorem:

$\underline{m=1}$

For $m=1$, one gets $\sum_{\substack{k_1 + \dots + k_m = n \\ k_1, \dots, k_m \geq 0}} \binom{n}{k_1, \dots, k_m} \prod_{t=1}^{m} \left(x^{(t)}\right)^{k_t} = \underbrace{\binom{n}{n}}_{= \frac{n!}{n!} = 1} \left(x^{(1)}\right)^n = \left(x^{(1)}\right)^n$

$\underline{m}$

Let one show that the theorem being satisfied for $m-1$ is also satisfied for m.

Well,

$$\left(x_1 + \dots + x_{m-1} + x_m\right)^n = \left(x_1 + \dots + (x_{m-1} + x_m)\right)^n (=) \longrightarrow$$

$$\longleftarrow (=) \left(x^{(1)} + \dots + x^{(m)}\right)^n = \sum_{\substack{k_1 + \dots + k_{m-2} + k = n \\ k_1, \dots, k_{m-2}, k \geq 0}} \binom{n}{k_1, \dots, k_{m-2}, k} \left(x^{(1)}\right)^{k_1} \dots \left(x^{(m-2)}\right)^{k_{m-2}} \left(x^{(m-1)} + x^{(m)}\right)^k =$$

(Multinomial theorem for $m-1$ terms in which the $m-1$-th term corresponds to $\left(x^{(m-1)} + x^{(m)}\right)$)

$$= \sum_{\substack{k_1 + \dots + k_{m-2} + k = n \\ k_1, \dots, k_{m-2}, k \geq 0}} \binom{n}{k_1, \dots, k_{m-2}, k} \left(x^{(1)}\right)^{k_1} \dots \left(x^{(m-2)}\right)^{k_{m-2}} \sum_{t=0}^{k} \binom{k}{t} \left(x^{(m-1)}\right)^t \left(x^{(m)}\right)^{k-t} =$$

(Binomial theorem applied to the last term)

$$= \sum_{t=0}^{k} \frac{k!}{t!(k-t)!} \left(x^{(m-1)}\right)^t \left(x^{(m)}\right)^{k-t} =$$

$$= \sum_{\substack{k_{m-1} + k_m = k \\ k_{m-1}, k_m \geq 0}} \binom{k}{k_{m-1}, k_m} \left(x^{(m-1)}\right)^{k_{m-1}} \left(x^{(m)}\right)^{k_m}$$

$t \longleftrightarrow k_{m-1}$

$k-t \longleftrightarrow k_m$

$$= \sum_{\substack{k_1 + \dots + k_{m-2} + k = n \\ k_1, \dots, k_{m-2}, k \geq 0}} \sum_{\substack{k_{m-1} + k_m = k \\ k_{m-1}, k_m \geq 0}} \binom{n}{k_1, \dots, k_{m-2}, k} \binom{k}{k_{m-1}, k_m} \left(x^{(1)}\right)^{k_1} \dots \left(x^{(m)}\right)^{k_m} (=)$$

$$\left(\binom{n}{k_1, \dots, k_{m-2}, k} \binom{k}{k_{m-1}, k_m} = \frac{n!}{k_1! \dots k_{m-2}! k!} \cdot \frac{k!}{k_{m-1}! k_m!} = \frac{n!}{k_1! \dots k_m!} = \binom{n}{k_1, \dots, k_m} \right) \longrightarrow$$

$$(\Rightarrow) \quad \left(x^{(1)} + \dots + x^{(m)}\right)^m = \sum_{\substack{k_1 + \dots + k_m = m \\ k_1, \dots, k_m \geq 0}} \binom{m}{k_1, \dots k_m} \left(x^{(1)}\right)^{k_1} \dots \left(x^{(m)}\right)^{k_m} \qquad \square$$

Note

(Binomial theorem)

Theorem:

For any non-negative integer m, the m-th power of the sum of the numbers x and y corresponds to

$$\boxed{(x+y)^m = \sum_{k=0}^{m} \binom{m}{k} x^k \, y^{m-k}} \tag{39}$$

where $\boxed{\binom{m}{k} = \frac{m!}{k!\,(m-k)!}}$ is a binomial coefficient. (40)

Demonstration:

The theorem may be proven through induction:

$\underline{m=1}$

$$\sum_{k=0}^{1} \binom{1}{k} x^k y^{1-k} = \binom{1}{0} x^0 y^1 + \binom{1}{1} x^1 y^0 = y + x = x + y$$

$\underline{m}$

Lets one show that the theorem being satisfied for $m-1$ is also satisfied for m.

Well,

→

←

$$(x+y)^m = (x+y)(x+y)^{m-1} \overset{\text{(Binomial theorem for the case } m-1)}{=} (x+y) \sum_{k=0}^{m-1} \binom{m-1}{k} x^k y^{m-1-k} =$$

$$= \left(\sum_{k=0}^{m-1} \binom{m-1}{k} x^{k+1} y^{m-(k+1)} \right) + \left(\sum_{k=0}^{m-1} \binom{m-1}{k} x^k y^{m-k} \right) =$$

$$= \left(\sum_{k=1}^{m} \binom{m-1}{k-1} x^k y^{m-k} \right) + \left(\sum_{k=0}^{m-1} \binom{m-1}{k} x^k y^{m-k} \right) =$$

$$= \sum_{k=1}^{m-1} \left(\binom{m-1}{k-1} + \binom{m-1}{k} \right) x^k y^{m-k} + \underbrace{\binom{m-1}{0}}_{=\binom{m}{0}} x^0 y^m + \underbrace{\binom{m-1}{m-1}}_{=\binom{m}{m}} x^m y^0 \quad (=1)$$

$$\left(\binom{m-1}{k-1} + \binom{m-1}{k} = \frac{(m-1)!}{(k-1)!\,(m-k)!} + \frac{(m-1)!}{k!\,(m-1-k)!} = \right.$$

$$= \frac{k(m-1)!}{k!\,(m-k)!} + \frac{(m-k)(m-1)!}{k!\,(m-k)!} = \frac{m(m-1)!}{k!\,(m-k)!} = \frac{m!}{k!\,(m-k)!} =$$

$$\left. = \binom{m}{k} \right)$$

→

$$\Rightarrow (x+y)^m = \left(\sum_{k=1}^{m-1} \binom{m}{k} x^k y^{m-k}\right) + \binom{m}{0} x^0 y^m + \binom{m}{m} x^m y^0$$

$$\Rightarrow (x+y)^m = \sum_{k=0}^{m} \binom{m}{k} x^k y^{m-k} \quad \square$$

←

The non-homogeneous polynomial kernel

Let $\{x\}, \{z\} \in \mathbb{R}^D$ and $\theta \in \mathbb{R}$. The non-homogeneous polynomial kernel is defined as

$$\boxed{K(\{x\},\{z\}) = (\{x\}^T\{z\} + \theta)^d, \text{ with } d \in \mathbb{N}^+} \quad (41)$$

(Non-homogeneous polynomial kernel)

Note that

$$K(\{x\},\{z\}) = (\{x\}^T\{z\} + \theta)^d = (x^{(1)}z^{(1)} + \dots + x^{(D)}z^{(D)} + \sqrt{\theta}\sqrt{\theta})^d =$$

$$= (\{x'\}\{z'\})^d = \leftarrow \text{(Multinomial theorem)}$$

$$\left(\text{Let } \{x'\} = \{x^{(1)}, \dots, x^{(D)}, \sqrt{\theta}\}, \quad \{z'\} = \{z^{(1)}, \dots, z^{(D)}, \sqrt{\theta}\}\right)$$

$$= \sum_{\substack{m_1+\dots+m_D+m_{D+1}=d \\ m_1,\dots,m_D,m_{D+1}\geq 0}} \binom{d}{m_1,\dots,m_D,m_{D+1}} (x^{(1)}z^{(1)})^{m_1} \dots (x^{(D)}z^{(D)})^{m_D} \sqrt{\theta}^{m_{D+1}} \sqrt{\theta}^{m_{D+1}} \quad (42)$$

$$\Rightarrow K(\{x\},\{z\}) = \sum_{\substack{m_1+\dots+m_D+m_{D+1}=d \\ m_1,\dots,m_D,m_{D+1}\geq 0}} \underbrace{\sqrt{\binom{d}{m_1,\dots,m_D,m_{D+1}}} (x^{(1)})^{m_1} \dots (x^{(D)})^{m_D} \sqrt{\theta}^{m_{D+1}}}_{=\varphi_j(\{x\})} \times$$

$$\times \underbrace{\sqrt{\binom{d}{m_1,\dots,m_D,m_{D+1}}} (z^{(1)})^{m_1} \dots (z^{(D)})^{m_D} \sqrt{\theta}^{m_{D+1}}}_{=\varphi_j(\{z\})} \Rightarrow$$

$$= \{\varphi(\{x\})\}^T \{\varphi(\{z\})\}.$$

The vector $\{\varphi(\{x\})\}$ may be expressed as

$$\boxed{\{\varphi(\{x\})\} = \left\{\sqrt{\binom{d}{m_1,\dots,m_D,m_{D+1}}} (x^{(1)})^{m_1} \dots (x^{(D)})^{m_D} \sqrt{\theta}^{m_{D+1}}\right\}_{\substack{m_1+\dots+m_D+m_{D+1}=d \\ m_1,\dots,m_D\geq 0}}} \quad (42)$$

(Mapping function associated with the non-homogeneous polynomial kernel)

And since there is an extra exponent, m_{D+1}, the dimensionality of $\{\varphi(\{x\})\}$ is (36) with D substituted by D+1:

$$\boxed{\frac{(d+D)!}{d!\,D!} = \binom{d+D}{d} = C_d^{d+D}} \quad (43)$$

(Dimensionality of the mapping function associated with the non-homogeneous polynomial kernel)

The radial basis function (RBF) Kernel

Let $\{x\}, \{z\} \in \mathbb{R}^D$ and $\sigma \in \mathbb{R}_0^+$. The RBF kernel is defined as

$$K(\{x\},\{z\}) = e^{-\frac{\|\{x\}-\{z\}\|^2}{2\sigma^2}} \tag{44}$$

(Radial basis function (RBF) Kernel)

Well,

$$K(\{x\},\{z\}) = e^{-\frac{\|\{x\}-\{z\}\|^2}{2\sigma^2}} = e^{-\frac{(\{x\}-\{z\})^T(\{x\}-\{z\})}{2\sigma^2}} =$$

$$= e^{-\frac{\|\{x\}\|^2}{2\sigma^2} - \frac{\|\{z\}\|^2}{2\sigma^2} + \frac{\{x\}^T\{z\}}{\sigma^2}} =$$

$$= e^{-\frac{\|\{x\}\|^2}{2\sigma^2}} \cdot e^{-\frac{\|\{z\}\|^2}{2\sigma^2}} \cdot e^{\frac{\{x\}^T\{z\}}{\sigma^2}} =$$

$$= \sum_{K=0}^{\infty} \frac{\left(\frac{\{x\}^T\{z\}}{\sigma^2}\right)^K}{K!} \cdot e^{-\frac{\|\{x\}\|^2}{2\sigma^2}} \cdot e^{-\frac{\|\{z\}\|^2}{2\sigma^2}} =$$

(Taylor series expansion of $e^{\frac{\{x\}^T\{z\}}{\sigma^2}}$ around $K = \frac{\{x\}^T\{z\}}{\sigma^2} = 0$)

$$= \sum_{K=0}^{\infty} \left(\frac{e^{-\frac{\|\{x\}\|^2}{2K\sigma^2}}}{\sigma\left(\sqrt{K!}\right)^{\frac{1}{K}}} \cdot \frac{e^{-\frac{\|\{z\}\|^2}{2K\sigma^2}}}{\sigma\left(\sqrt{K!}\right)^{\frac{1}{K}}} \cdot \{x\}^T\{z\} \right)^K \quad (=$$

$$(= K(\{x\},\{z\}) = \sum_{K=0}^{\infty} \left(\sum_{i=0}^{D} \frac{e^{-\frac{\|\{x\}\|^2}{2K\sigma^2}}}{\sigma\left(\sqrt{K!}\right)^{\frac{1}{K}}} x^{(i)} \cdot \frac{e^{-\frac{\|\{z\}\|^2}{2K\sigma^2}}}{\sigma\left(\sqrt{K!}\right)^{\frac{1}{K}}} z^{(i)} \right)^K \quad (=$$

$$(\Rightarrow K(\{x\},\{z\}) = \sum_{K=0}^{\infty} \sum_{\substack{m_1+\cdots+m_D=K \\ m_1,\ldots,m_D \geq 0}} \left(\underbrace{\frac{e^{-\frac{\|\{x\}\|^2}{2K\sigma^2}}}{\sigma\left(\sqrt{K!}\right)^{\frac{1}{K}}} \sqrt{\binom{K}{m_1,\ldots,m_D}} \left(x^{(1)}\right)^{m_1} \cdots \left(x^{(D)}\right)^{m_D}}_{=\varphi_{\vec{m}}(\{x\})} \times \right.$$

(Multinomial theorem)

$$\left. \times \underbrace{\frac{e^{-\frac{\|\{z\}\|^2}{2K\sigma}}}{\sigma\left(\sqrt{K!}\right)^{\frac{1}{K}}} \sqrt{\binom{K}{m_1,\ldots,m_D}} \left(z^{(1)}\right)^{m_1} \cdots \left(z^{(D)}\right)^{m_D}}_{=\varphi_{\vec{m}}(\{z\})} \right) =$$

$$= \{\varphi(\{x\})\}^T \{\varphi(\{z\})\}.$$

The vector $\{\varphi(\{x\})\}^T$ may be expressed as

$$\{\varphi(\{x\})\} = \left\{ \frac{e^{-\frac{\|\{x\}\|^2}{2K\sigma^2}}}{\sigma\left(\sqrt{K!}\right)^{\frac{1}{K}}} \cdot \sqrt{\binom{K}{m_1,\ldots,m_D}} \left(x^{(1)}\right)^{m_1} \cdots \left(x^{(D)}\right)^{m_D} \right\}_{\substack{K=0,\ldots,\infty \\ m_1+\cdots+m_D=K \\ m_1,\ldots,m_D \geq 0}} \tag{45}$$

(Mapping function associated with the RBF Kernel)

The dimensionality of the RBF kernel is ∞.

Multiclass classification

The theory enunciated in this chapter is with respect to a binary (two classes) classification. Still, one may consider a procedure to extend it to a multiclass one, termed "one versus rest". Let q be the number of classes. The procedure is such that the multiclass problem is transformed into q binary classification problems.

Let $\{C\} = \{C_1, \dots, C_q\}$ be the set of labels associated with the respective q classes. A SVM model for the j-th class is defined by first giving the value 1 to the labels y of the data points associated to that class and 0 to all the other ones. Next, a decision boundary is obtained through the SVM methodology for a binary class. Let $\{w_j\}$ and b_j be the respective parameters. All classes would then have their respective decision boundary and a binary SVM model. The multiclass SVM model would then be such that the chosen label for the feature $\{x\}$ is the one for wich the respective decision boundary is the closest to that point being the displacement of the hyperplane that passes through $\{x\}$ from the decision boundary positive (as that would positively associate the label to it in the binary SVM model).

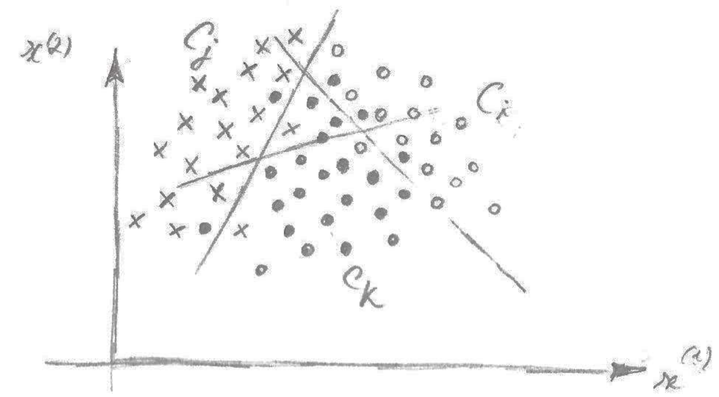
The displacement of the hyperplane that passes through $\{x\}$ from the decision boundary associated with the j-th class is according to (6) and (2),

$$\boxed{d_j = \frac{\{w_j\}^T \{x\} - b_j}{\|\{w_j\}\|}}. \tag{46}$$

The label to be given to the feature $\{x\}$ would then be

$$\boxed{C(\{x\}) = C_j : d_j = \min\left(\{d_k \; \forall \; k=1,\dots,q : d_k > 0\}\right)}. \tag{47}$$

(Multiclass SVM model)



If there is no $d_k > 0 \; \forall k = 1, \dots, q$, then, the label C_j associated with the minimum d_k value is picked.

I- Bayesian Decision Theory

Joint probability Distributions

Let one consider as an example measurements of Body Mass Index (BMI), X = weight / height^2 of several people and the association of a respective score, H, which corresponds either to h_1 - standing for "person being over-weight" - or h_2 - standing for "person being of normal weight". Also, let one consider that the measurements of X are such that it may correspond to K different values - $x_1, x_2, \dots x_K$.

If K=5, one may for instance get the joint probability table presented below:

X \ H	h_1	h_2	$P(x_i) = \sum_j P(x_i, h_j)$ (X-marginal probability)
x_1	$P(x_1, h_1) = 2/22$	0/22	2/22
x_2	5/22	0/22	5/22
x_3	4/22	3/22	7/22
x_4	2/22	3/22	5/22
x_5	1/22	2/22	3/22
$P(h_j) = \sum_i P(x_i, h_j)$ (H-marginal probability)	14/22	8/22	$\sum_j \sum_i P(x_i, h_j) = \sum_i P(x_i) = \sum_j P(h_j) = 1$

The (i,j)-th entry of the table corresponds to the probability of the measurement giving values $X = x_i$ and $H = h_j$ - the ratio between the number of training examples with such values and the total number of training examples, assuming that the training examples are i.i.d. (independent and identically distributed). This is the so-called joint probability,

$$\boxed{P(x_i, h_j) := P(X = x_i, H = h_j)} \quad (48)$$

(joint probability)

In the margins of the table, we have the result of the so-called marginalizations: the right margin presents the marginal probability distribution for X,

$$\boxed{P(x_i) = \sum_j P(x_i, h_j)}, \quad (49)$$

(Marginal probability distribution for X)
(or evidence)

and the bottom margin presents the marginal probability distribution for H,

$$\boxed{P(h_j) = \sum_i P(x_i, h_j)}. \quad (50)$$

(Marginal probability distribution for H)
(or prior)

The probabilities $P(h_j)$ are the so-called priors - the probabilities of a person being over-weight (or not) before we see any measurement. The probabilities $P(x_i)$ are the so-called evidences - the probabilities of receiving such BMI measurements to begin with.

Note that,

$$\boxed{\sum_i P(x_i) = \sum_j P(h_j) = \sum_{ij} P(x_i, h_j) = 1}. \quad (51)$$

The conditional probability of $H = h_j$ given that $X = x_i$ corresponds to the ratio between the number of training examples with $(X, H) = (x_i, h_j)$ and the number of training examples with $X = x_i$, that is,

$$\boxed{P(h_j | x_i) = \frac{P(x_i, h_j)}{P(x_i)}} \quad (52)$$

(Conditional probability of $H = h_j$ given that $X = x_i$)
(or posterior probability)

This probability is also termed posterior probability. The conditional probability of $X = x_i$ given that $H = h_j$, also termed class conditional likelihood, corresponds to

$$\boxed{P(x_i | h_j) = \frac{P(x_i, h_j)}{P(h_j)}} \quad (53)$$

(Conditional probability of $X = x_i$ given that $H = h_j$)
(or class conditional likelihood)

Note also that

$$\sum_j P(h_j|x_i) \overset{(52)}{=} \sum_j \frac{P(x_i,h_j)}{P(x_i)} = \frac{\sum_j P(x_i,h_j)}{P(x_i)} \overset{(49)}{=} \frac{P(x_i)}{P(x_i)} = 1 \qquad (54)$$

Likewise,

$$\sum_i P(x_i|h_j) = 1 \qquad (55)$$

There is a way to compute the posterior probability through the prior, evidence and class conditional likelihood:

$$(52): \quad P(h_j|x_i) = \frac{P(x_i,h_j)}{P(x_i)} \overset{(53)}{=} \frac{P(x_i|h_j)\,P(h_j)}{P(x_i)}$$

$$P(h_j|x_i) = \frac{P(x_i|h_j)\,P(h_j)}{P(x_i)} \qquad (56)$$

(Bayes' theorem)

"Proper Bayes" decision principle

Let $l(h_i,h_j)$ be the loss incurred by assigning the class h_i for $[x]$ when in fact h_j is the correct class. One typical loss function, is the so-called "0/1" one:

$$l(h_i,h_j) = \begin{cases} 0 & \text{if } i=j \\ 1 & \text{if } i\neq j \end{cases} = 1-\delta_{ij} \qquad (57)$$

("0/1" loss function)

One may define the expected risk, which is defined as the expected value for the loss when labelling with h_i the given feature $X=x$:

$$R(h_i|x) = \sum_j l(h_i,h_j)\,P(h_j|x) \qquad (58)$$

(Expected risk of labelling with h_i the given feature $X=x$)

The "proper Bayes" decision when assigning a label H for the given feature $X=x$ would be one that minimizes the respective expected risk. Let h^* be this optimum label. It then corresponds to

$$h^* = \underset{h_i}{\operatorname{argmin}}\, R(h_i|x) \qquad (59)$$

(Optimum label for given feature $X=x$, according to a "proper Bayes" decision principle)

Maximum A Posteriori (MAP) decision principle

In the case of the loss function corresponding to the "0/1" one, (57) the "proper Bayes" decision principle is reduced to the so-called Maximum A Posteriori decision principle:

$$(59): \quad h^* = \underset{h_i}{\operatorname{argmin}}\, R(h_i|x) \overset{(58)}{=} \underset{h_i}{\operatorname{argmin}} \sum_j l(h_i,h_j)\,P(h_j|x) =$$

$$(\Rightarrow) \quad h^* = \underset{h_i}{\operatorname{argmin}} \sum_j (1-\delta_{ij}) P(h_j|x) =$$

$$= \underset{h_i}{\operatorname{argmin}} \left(\underbrace{\sum_j P(h_j|x)}_{=1 \text{ according to (54)}} - \sum_j \delta_{ij} P(h_j|x) \right) =$$

$$= \underset{h_i}{\operatorname{argmin}} \; 1 - P(h_i|x) \quad (\Leftarrow \text{ minimizing } 1-y \text{ is equivalent to maximize } y)$$

$$(\Leftarrow) \quad \boxed{h^* = \underset{h_i}{\operatorname{argmax}} \; P(h_i|x)} \qquad (60)$$

(Optimum label for given feature $X = x$ according to the Maximum A Posteriori decision principle that holds if the loss function is a "0/1" one.)

Maximum Likelihood (ML) decision principle

In the case of the loss function being the "0/1" one, and the prior $P(h)$ being uniform, the Maximum A Posteriori decision principle is reduced to the Maximum Likelihood decision principle:

$$(60): \; h^* = \underset{h_i}{\operatorname{argmax}} \; P(h_i|x) \overset{(56)}{=} \underset{h_i}{\operatorname{argmax}} \; \frac{P(x|h_i)P(h_i)}{P(x)} =$$

$$\overset{\nearrow}{=} \underset{h_i}{\operatorname{argmax}} \; P(x|h_i)P(h_i) \quad (\Leftarrow)$$

($P(x)$ does not depend on h_i)

($P(h_i)$ is the same for all h_i; it does then not depend on h_i)

$$(\Leftarrow) \quad \boxed{h^* = \underset{h_i}{\operatorname{argmax}} \; P(x|h_i)} \qquad (61)$$

(Optimum label for given feature $X = x$ according to the Maximum Likelihood decision principle that holds if the loss function is a "0/1" one and the prior $P(h)$ is uniform)

Note that in the case of the Maximum A Posteriori principle the optimum label is the one that maximizes the posterior probability. As in the case of the Maximum Likelihood principle, the optim label is the one that maximizes the class conditional likelihood.

Two statistically independent variables

Let one consider two statistically independent variables X and Y. Symbolically, the independence is expressed as

$$X \perp Y.$$

Since X and Y is independent the information about the value of Y does not anything about X, and, therefore,

$$\boxed{P(X|Y) = P(X)} \qquad (62)$$

It is possible to show that (52) implies $P(X,Y)=P(X)P(Y)$:

$$P(X,Y) \overset{(52)}{=} P(X|Y)\,P(Y) \overset{(62)}{\Rightarrow} \boxed{P(X,Y)=P(X)P(Y)} \quad (63)$$

(Joint probability for the case of X and Y being independent)

The independency condition is quite useful in reducing the number of required terms to fully describe a joint probability table. Let X and Y range over k and l distinct values, respectively. In the case of X and Y not being independent, the joint probabilities $P(x_i,y_j)$ would be obtained from each one of the $k\cdot l$ members except one since they have to satisfy the constraint $\sum P(x_i,y_j)=1$.

The total number of required terms is $k\cdot l-1$.

[**Note:** (Constraints associated with the range $[0,1]$ of the joint probabilities

Actually the number of required terms to fully define the joint probabilities $P(x_i,y_j)$ in the case of X and Y not being independent is less than $k\cdot l-1$ since one needs to account the constraints associated with the range $[0,1]$ to which the joint probabilities pertain. However for the sake of simplicity such constraints are unaccounted here.]

If X and Y are independent then the joint probabilities $P(x_i,y_j)$ would be fully determined by the marginal probabilities $P(x_i)$ and $P(y_j)$. The number of required terms is $k+l-2$ (one needs to take into

account two independent constraints: $\sum_i P(x_i)=1$, $\sum_j P(y_j)=1$.

Many statistically independent variables

Let one consider the case of n statistically independent variables: $X^{(1)}, X^{(2)}, \dots, X^{(n)}$. That is,

$$X^{(1)} \perp X^{(2)} \perp \dots \perp X^{(n)}.$$

Such condition implies that

$$\boxed{P(X^{(i)}|X^{(j)})=P(X^{(i)})\ \forall i,j=1,\dots,n \wedge i\neq j}. \quad (64)$$

Well,

$$P(X^{(1)},\dots,X^{(n)}) \overset{(64)}{=} P(X^{(1)}|X^{(2)},\dots,X^{(n)})\,P(X^{(2)},\dots,X^{(n)}) \overset{(52)}{=}$$

$$\overset{(52)}{=} P(X^{(1)})\,P(X^{(2)}|X^{(3)}\dots X^{(n)})P(X^{(3)},\dots,X^{(n)}) \overset{(52)}{=}$$

$$= \dots \Rightarrow$$

$$\Rightarrow \boxed{P(X^{(1)},\dots,X^{(n)})=P(X^{(1)})\times\dots\times P(X^{(n)})}. \quad (65)$$

(Joint probability for the case of $X^{(1)},\dots,X^{(n)}$ being independent)

The number of required terms for fully describing the joint probabilities $P(x_i^{(1)},\dots,x_j^{(n)})$ is $K_1+\dots+K_n-n$, where K_q is the number of distinct values that the variable $X^{(q)}$ may take. The subtraction of n is due to the constraints $\sum_i P(x_i^{(q)})=1\ \forall q=1,\dots,n$.

Conditional independence

Let one consider the case of two variables, X and Y, being independent given another, H. Such conditional independence is expressed as

$$X \perp Y \mid H.$$

Under such conditions, one gets

$$\boxed{P(X \mid Y, H) = P(X \mid H)} \quad (66)$$

Therefore

$$P(X,Y \mid H) \underset{(52)}{=} P(X \mid Y, H)\, P(Y,H) \quad (66)$$

$$\boxed{P(X,Y \mid H) = P(X \mid H) \cdot P(Y \mid H)} \quad (67)$$

(Conditional joint probability for the case of X and Y being conditionally independent)

Let K, r and s be the numbers of indistinct values that the variables X, Y and Y may take, respectively. If X and Y were not conditionally independent, then one would need $K \times r \times s - 1$ terms to fully describe the joint probabilities $P(x_i, y_j, h_l)$. In the case of X and Y

being conditionally independent one has from (52) and (67) that

$$\boxed{P(X,Y,H) = P(X \mid H) \cdot P(Y \mid H) \cdot P(H)}. \quad (68)$$

Therefore the number of required terms to fully describe the joint probabilities $P(x_i, y_i, h_l)$ would be $s \times (K + r - 2)$. The subtraction by 2 is due to the constraints $\sum_i P(x_i \mid H) = 1$ and $\sum_j P(y_j \mid H) = 1$.

An example of conditional independence is the case of two persons, let these be Joe and Mo, living on opposites sides of a city and commuting to work through train and car, respectively. Let X be the event "Joe is late for work" and Y be the event "Mo is late for work". Due to particular factors, X and Y may not be independent. For instance if there was a train strike that would cause Joe to be late, it could also cause Mo to be late since more people would be using their car instead of the train creating extra traffic. Therefore one should consider a third variable, H, standing for the event "train strike" which would conditionally decouple X from Y.

Example: Estimation of a bias of a coin

Let one consider the case of coin tossing, in which "heads" is associated with the value $X=1$ and "tails" is associated with the value $X=0$. Also, let one consider that there is a coin bias: the probability of having "heads" in the tossing, q, which is a parameter of the experience. Therefore,

$$\boxed{P(X=1|H=q)=q, \text{ and } P(X=0|H=q)=1-q}, \tag{69}$$

where $H=q$ is the hypothesis for the bias. For the coin to be fair the bias would need to be $q=0.5$. Let one estimate q using the Maximum Likelihood principle and considering an experiment of m tossings. Let $x_1, \ldots, x_m$ be the values of the X variable associated with the tossings. Well, assuming the prior $P(H)$ to be uniform, one would have from the Maximum Likelihood principle that

$$\boxed{q^* = \arg\max_q P(x_1, \ldots, x_m | H=q)} \tag{70}$$

In (70), $P(x_1, \ldots, x_m | H=q)$ corresponds to the probability that $X_1=x_1, \ldots$ and $X_m=x_m$ in the respective m tossings given the bias $H=q$. Since the tossings are i.i.d., $X_1, \ldots, X_m | H$, and (70) and (68) one would have

$$q^* = \arg\max_q \prod_{i=1}^{m} P(x_i|q) \overset{\text{ln}}{=} \boxed{q^* = \arg\max_q \sum_{i=1}^{m} \ln P(x_i|q)} \tag{71}$$

(It is easier to handle sums than products. The logarithm of a product is a sum of logarithms and finding the maximum of a function is equivalent to finding the maximum of the logarithm of a function due to the fact that the logarithm is a monotonously increasing function.)

[**Note:** (Monotonicity of the logarithmic function)

It may be shown that the logarithm of a function has the same monotony as the function. For the logarithmic function to be valid the argument x needs to be positive, that is, $x>0$. Well,

$$\frac{\partial}{\partial x} \ln f(x) = \frac{f'(x)}{f(x)}$$

$$\text{sgn}\left(\frac{\partial}{\partial x} \ln f(x)\right) = \text{sgn } f'(x) \cdot \underbrace{\text{sgn } f(x)}_{>0} = \text{sgn } f'(x). \quad \square$$

Let $\lambda = |\{x_i = 1 \ \forall i = 1, \dots, m\}|$ be the number of tossings in which "heads" was obtained. Therefore,

$$(71): \quad q^* = \underset{q}{\operatorname{argmax}} \ \lambda \log \underbrace{P(1|q)}_{=q} + (m-\lambda) \log \underbrace{P(0|q)}_{=1-q} \Leftrightarrow$$

$$\Leftrightarrow \boxed{q^* = \underset{q}{\operatorname{argmax}} \ \lambda \log q + (m-\lambda) \log(1-q)} \quad (72)$$

Well,

$$\frac{\partial}{\partial q}\left(\lambda \log q + (m-\lambda)\log(1-q)\right)\Big|_{q=q^*} = 0 \Leftrightarrow$$

$$\Leftrightarrow \frac{\lambda}{q^*} - \frac{m-\lambda}{1-q^*} = 0 \Leftrightarrow \frac{(1-q^*)\lambda - (m-\lambda)q^*}{q^*(1-q^*)} = 0 \Leftrightarrow$$

$$\Leftrightarrow \lambda - m q^* = 0 \ \wedge \ q^* \notin \{0,1\} \Leftrightarrow$$

$$\Leftrightarrow \boxed{q^* = \frac{\lambda}{m}} \quad (73)$$

(If $\lambda = 0$, $q^* = \operatorname{argmax} m \log(1-q) = 0$
If $\lambda = m$, $q^* = \operatorname{argmax} m \log q = 1$.
Both conditions are satisfied by $q^* = \frac{\lambda}{m}$)

(Bias of a coin m-times tossed coin in which "heads" occurred λ times — probability of "heads" coming out according to the Maximum Likelihood principle)

Note that

$$\frac{\partial^2}{\partial q^2}\left(\lambda \log q + (m-\lambda)\log(1-q)\right)\Big|_{q=q^*} =$$

$$= \frac{\partial}{\partial q}\left(\frac{\lambda}{q} - \frac{m-\lambda}{1-q}\right)\Big|_{q=q^*} = -\frac{\lambda}{q^{*2}} - \frac{m-\lambda}{(1-q^*)^2} =$$

$$\overset{(73)}{=} -\frac{\lambda}{\frac{\lambda^2}{m^2}} + \frac{m-\lambda}{\left(1-\frac{\lambda}{m}\right)^2} = -\frac{m^2}{\lambda} - \frac{m-\lambda}{\frac{(m-\lambda)^2}{m^2}} =$$

$$= -\frac{m^2}{\lambda} - \frac{m^2}{m-\lambda} = \frac{-(m-\lambda)m^2 - \lambda m^2}{\lambda(m-\lambda)} = -\frac{m^3}{(m-\lambda)\lambda} \quad (*)$$

$$\Leftrightarrow \frac{\partial^2}{\partial q^2}\left(\lambda \log q + (m-\lambda)\log(1-q)\right)\Big|_{q=q^*} < 0 \ \text{ if } \ \lambda \in]0, m[$$

If $\lambda = 0$,

$$\frac{\partial^2}{\partial q^2}\left(\lambda \log q + (m-\lambda)\log(1-q)\right)\Big|_{q=q^*} = \frac{\partial^2}{\partial q^2}\left(m \log(1-q)\right)\Big|_{q=q^*} =$$

$$= \frac{\partial}{\partial q}\left(-m\frac{1}{1-q}\right)\Big|_{q=q^*} = -\frac{m}{(1-q^*)^2} \overset{(73): q^* = \frac{\lambda}{m} = 0}{=} -m < 0$$

If $\lambda = m$,

$$\frac{\partial^2}{\partial q^2}\left(\lambda \log q + (m-\lambda)\log(1-q)\right)\Big|_{q=q^*} = \frac{\partial^2}{\partial q^2}\left(m \log q\right)\Big|_{q=q^*} \Leftrightarrow$$

$$\Leftrightarrow \left.\frac{\partial^2}{\partial q^2}\left(n \lg q\right)\right|_{q=q^*} = \left.\frac{\partial}{\partial q}\left(\frac{n}{q}\right)\right|_{q=q^*} = -\frac{n}{q^{*2}} = -m < 0$$

$$(73): \; q^* = \frac{1}{m} = 1$$

Since $\left.\frac{\partial^2}{\partial q^2}\left(\lambda \lg q + (m-\lambda)\lg(1-q)\right)\right|_{q=q^*} < 0 \quad \forall \lambda \in [0, m]$,

one may conclude that $q^* = \frac{1}{m}$, as given by (73) maximises $P(x_1, \dots, x_m | q)$.

Example: estimation of Gaussian probability density function

Let one consider an i.i.d. sample of K points, $S = \{\{x_1\}, \dots, \{x_K\}\}$ with $\{x_i\} \in \mathbb{R}^n \; \forall i = 1, \dots, m$. It would be important to determine the Gaussian probability density function that is according to the Maximum Likelihood principle the most appropriate one for describing the sample. Note, however, that a conditional probability density function of variable $\{X\}$ and hypothesis $\{H\}$ (set of parameters), let this be $f(\{x\}|\{H\})$, may be solely regarded for the case of $\{X\}$ and $\{H\}$ being continuous. The Maximum Likelihood principle (64) should be reexpressed for that case.

Note that $f(\{x\}|\{H\})$ is such that $f(\{x\}|\{H\})\, d\{x\} = f(\{x\}|\{H\})\, dx_1 \dots dx_n$ gives the probability

$$P(\{X\} \in [x_1, x_1 + dx_1] \times \dots [x_n, x_n + dx_n] \,|\, \{H\}).$$

Being $\{X\}$ and $\{H\}$ continuous variables, the expected risk which corresponds to the expected value for the loss when labelling with $\{h\}$ the given feature $\{X\} = \{x\}$ is given by

$$R(\{h\}|\{x\}) = \int l(\{h\}, \{h'\})\, f(\{x\}|\{H\})\, d\{h'\} \tag{74}$$

(Expected risk of labelling with $\{h\}$ the given feature $\{X\} = \{x\}$ in the case of $\{X\}$ and $\{H\}$ being continuous)

where $l(\{h\}, \{h'\})$ is the loss function. According to the "proper Bayes" decision principle, the optimum label $\{h^*\}$ would be the one that minimises the respective expected risk:

$$\{h^*\} = \underset{\{h\}}{\operatorname{argmin}}\, R(\{h\}|\{x\}) \tag{75}$$

(Optimum label for given feature $\{X\} = \{x\}$, according to a "proper Bayes" decision principle, in the case of $\{X\}$ and $\{H\}$ being continuous)

Let one consider a "-os/x" loss function:

$$l(\{h\}, \{h'\}) = \begin{cases} -\infty \text{ if } \{h\} \neq \{h'\} \\ 1 \text{ if } \{h\} \neq \{h'\} \end{cases} = 1 - \delta(\{h\} - \{h'\}) \tag{76}$$

In that case the "proper Bayes" decision principle is reduced to the Maximum A Posteriori principle:

$$(75):\quad \{h^*\} = \underset{\{h\}}{\arg\min}\ R(\{h\}|\{x\}) =$$

(74) (76)

$$= \underset{\{h\}}{\arg\min} \int \left(1 - \delta(\{h\} - \{h'\})\right) p(\{h'\}|\{x\})\, d\{h'\} =$$

$$= \underset{\{h\}}{\arg\min} \left(\underbrace{\int p(\{h'\}|\{x\})\, d\{h'\}}_{=1} - \underbrace{\int \delta(\{h\} - \{h'\})\, p(\{h'\}|\{x\})\, d\{h'\}}_{= p(\{h\}|\{x\})} \right) =$$

$$= \underset{\{h\}}{\arg\min}\ 1 - p(\{h\}|\{x\}) \iff$$

$$\iff \boxed{\{h^*\} = \underset{\{h\}}{\arg\max}\ p(\{h\}|\{x\})} \quad (77)$$

(Optimum label for given feature $\{X\} = \{x\}$ according to the Maximum A Posteriori decision principle that holds if the loss function is a "$-\infty/1$" one, in the case of $\{X\}$ and $\{H\}$ being continuous)

Additionally, if the prior $p(\{h\})$ is uniform, the Maximum A Posteriori decision principle is reduced to the Maximum Likelihood decision principle:

$$(77):\quad \{h^*\} = \underset{\{h\}}{\arg\max}\ p(\{h\}|\{x\}) = \underset{\{h\}}{\arg\max}\ \frac{p(\{x\}|\{h\})\, p(\{h\})}{p(\{x\})}$$

(Analogous Bayes' theorem (56) for the case of probability density functions)

($p(\{x\})$ does not depend on $\{h\}$)

$\iff$

($p(\{h\})$ is uniform, not depending on $\{h\}$)

$$\boxed{\{h^*\} = \underset{\{h\}}{\arg\max}\ p(\{x\}|\{h\})} \quad (78)$$

(Optimum label for given feature $X = x$ according to the Maximum Likelihood principle that holds if the loss function is "$-\infty/1$" and $p(\{h\}$ is uniform, considering $\{X\}$ and $\{H\}$ to be continuous variables)

By going back to the current example, one may state that the proposed Gaussian probability density function is

$$\boxed{p(\{x\}|\{\theta\}) = \frac{e^{-\frac{1}{2}(\{x\}-\{\mu\})^T [\Sigma]^{-1} (\{x\}-\{\mu\})}}{\sqrt{(2\pi)^n\, |[\Sigma]|}}} \quad (79)$$

(Gaussian probability density function)

where $\{\mu\}$ is the mean vector, $[\Sigma]$ is the covariance matrix, $\{\theta\} = (\{\mu\}, [\Sigma])$ is the set of parameters of the probability density function and n is the dimension of vector $\{x\}$.

The Maximum Likelihood principle (78) applied to the set of samples $S = \{\{x_1\}, \dots, \{x_K\}\}$ corresponds to

$$\{\theta^*\} = \underset{\{\theta\}}{\operatorname{argmax}}\ f(\{x_1\}, \dots, \{x_K\} | \{\theta\}) = \left(\begin{array}{l}\text{The samples}\\ \{x_1\}, \dots, \{x_K\} \text{ are}\\ \text{i.i.d.}\end{array}\right)$$

$$= \underset{\{\theta\}}{\operatorname{argmax}} \prod_{i=1}^{K} f(\{x_i\} | \{\theta\}) = \left(\begin{array}{l}\text{Applying the}\\ \text{logarithmic}\\ \text{function}\end{array}\right)$$

$$= \underset{\{\theta\}}{\operatorname{argmax}} \sum_{i=1}^{K} \log\left(f(\{x_i\} | \{\theta\})\right) =$$

$$= \underset{\{\theta\}}{\operatorname{argmax}} \sum_{i=1}^{K} \log\left(\frac{e^{-\frac{1}{2}(\{x_i\}-\{\mu\})^T [\Sigma]^{-1} (\{x_i\}-\{\mu\})}}{\sqrt{(2\pi)^n |[\Sigma]|}}\right)$$

$$\overset{(*)}{=} \underset{\{\theta\}}{\operatorname{argmax}} \left(-\frac{K}{2}\log|[\Sigma]| - \frac{Kn}{2}\log(2\pi) - \sum_{i=1}^{K} \frac{1}{2}(\{x_i\}-\{\mu\})^T [\Sigma]^{-1} (\{x_i\}-\{\mu\})\right) (=)$$

$$(=) \boxed{\{\theta^*\} = \underset{\{\theta\}}{\operatorname{argmax}}\ \mathcal{L}(\{\theta\})}, \quad (80)$$

where

$$\boxed{\mathcal{L}(\{\theta\}) = -\frac{K}{2}\log|[\Sigma]| - \frac{Kn}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{K} (\{x_i\}-\{\mu\})^T [\Sigma]^{-1} (\{x_i\}-\{\mu\})}, \quad (81)$$

is the Likelihood Function.

Well,

$$\frac{\partial \mathcal{L}}{\partial \mu^{(j)}} = -\frac{1}{2}\sum_{i=1}^{K} \frac{\partial}{\partial \mu^{(j)}} \left((\{x_i\} - \{\mu\})^T \sum_{m=1}^{n}\sum_{l=1}^{n} ([\Sigma]^{-1})_{ml} (x_i^{(l)} - \mu^{(l)}) \{e_m\} \right) =$$

$$= -\frac{1}{2}\sum_{i=1}^{K} \frac{\partial}{\partial \mu^{(j)}} \left(\sum_{m=1}^{n}\sum_{l=1}^{n} ([\Sigma]^{-1})_{ml} (x_i^{(l)} - \mu^{(l)})(x_i^{(m)} - \mu^{(m)}) \right) =$$

$$= -\frac{1}{2}\sum_{i=1}^{K}\sum_{m=1}^{n}\sum_{l=1}^{n} ([\Sigma]^{-1})_{ml} \Big((-\delta_{jl})(x_i^{(m)} - \mu^{(m)}) + (-\delta_{jm})(x_i^{(l)} - \mu^{(l)}) \Big) =$$

$$= -\frac{1}{2}\sum_{i=1}^{K} \left(-\sum_{m=1}^{n} ([\Sigma]^{-1}_{mj})(x_i^{(m)} - \mu^{(m)}) - \sum_{l=1}^{n} ([\Sigma]^{-1}_{jl})(x_i^{(l)} - \mu^{(l)}) \right) =$$

$$= \frac{1}{2}\sum_{i=1}^{K} \left(\sum_{m=1}^{n} [\Sigma]^{-1,T}_{jm} (x_i^{(m)} - \mu^{(m)}) + \sum_{l=1}^{n} [\Sigma]^{-1}_{jl} (x_i^{(l)} - \mu^{(l)}) \right) =$$

$$= \frac{1}{2}\sum_{i=1}^{K} \Big(\{e_j\}^T [\Sigma]^{-1,T} (\{x_i\} - \{\mu\}) + [e_j]^T [\Sigma]^{-1} (\{x_i\} - \{\mu\}) \Big) (=)$$

$$(=) \boxed{\frac{\partial \mathcal{L}}{\partial \mu^{(j)}} = \{e_j\}^T \frac{1}{2}\left([\Sigma]^{-1,T} + [\Sigma]^{-1}\right)\left(\left(\sum_{i=1}^{K} \{x_i\}\right) - K\{\mu\}\right)} \quad (82)$$

$$\therefore \left.\frac{\partial \mathcal{L}}{\partial \mu^{(j)}}\right|_{\mu^{(j)} = \mu^{*(j)}} \overset{=0\ \forall j=1,\dots,n}{(=)} \{e_j\}^T \frac{1}{2}\left([\Sigma]^{-1,T} + [\Sigma]^{-1}\right)\left(\left(\sum_{i=1}^{K} \{x_i\}\right) - K\{\mu^*\}\right) = 0 \ \forall j=1,\dots,n \ (=)$$

$$(=) \ [I]\frac{1}{2}\left([\Sigma]^{-1,T} + [\Sigma]^{-1}\right)\left(\left(\sum_{i=1}^{K} \{x_i\}\right) - K\{\mu^*\}\right) = \{0\} \ (=)$$

$$\Rightarrow \boxed{\frac{1}{2}\left(\left[\Sigma\right]^{-1,T} + \left[\Sigma\right]^{-1}\right)\left(\left(\sum_{i=1}^{K} \{N_i\}\right) - K\{\mu^*\}\right) = \{0\}} \tag{83}$$

Note: (The covariance matrix $[\Sigma]$ is symmetric and positive semi-definite)

The (ij)-th term of the covariance matrix Σ_{ij} corresponds to the expected value of the multiplication between the displacement of the i-th variable from its mean $x^{(i)} - \mu^{(i)}$ and the displacement of the j-th variable from its mean $x^{(j)} - \mu^{(j)}$, that is,

$$\Sigma_{ij} = \mathbb{E}\left((x^{(i)} - \mu^{(i)}) \cdot (x^{(j)} - \mu^{(j)})\right) \Rightarrow$$

$$\Rightarrow \sum_{ij} \{e_i\}\{e_j\}^T = \mathbb{E}\left((x^{(i)} - \mu^{(i)})\{e_i\}(x^{(j)} - \mu^{(j)})\{e_j\}^T\right) \Rightarrow \left(\begin{array}{c}\text{The expected}\\ \text{value operator}\\ \text{is linear}\end{array}\right)$$

$$\Rightarrow \underbrace{\sum_{ij} \Sigma_{ij}\{e_i\}\{e_j\}^T}_{=[\Sigma]} = \mathbb{E}\left(\underbrace{\sum_i (x^{(i)} - \mu^{(i)})\{e_i\}}_{=\{x\} - \{\mu\}} \cdot \underbrace{\sum_j (x^{(j)} - \mu^{(j)})\{e_j\}^T}_{=(\{x\} - \{\mu\})^T}\right) \Rightarrow$$

$$\boxed{[\Sigma] = \mathbb{E}\left((\{x\} - \{\mu\})(\{x\} - \{\mu\})^T\right)}, \tag{84}$$

(Covariance matrix)

It is easy to conclude from (84) that $[\Sigma]$ is a symmetric matrix:

$$\boxed{[\Sigma]^T = [\Sigma]}. \tag{85}$$

(Covariance matrix is symmetric)

$\Leftarrow$ Also, one may show that the covariance matrix is positive semi-definite, that is,

$$\boxed{\{N\}^T[\Sigma]\{N\} \geq 0 \quad \forall \{N\} \text{ of appropriate size}} \tag{86}$$

Indeed, (The covariance matrix is positive semi-definite)

$$\{N\}^T[\Sigma]\{N\} \overset{(84)}{=} \{N\}^T \mathbb{E}\left((\{x\} - \{\mu\})(\{x\} - \{\mu\})^T\right)\{N\} = \left(\begin{array}{c}\text{The expected value}\\ \text{operator is linear}\end{array}\right)$$

$$= \mathbb{E}\left(\underbrace{\{N\}^T(\{x\} - \{\mu\})}_{=:s}(\{x\} - \{\mu\})^T\{N\}\right) =$$

$$= \mathbb{E}(s^2) \geq 0 \text{ since } s \text{ is a real number and the square of any real number is non-negative}$$

Note: (A matrix with a null eigenvalue is not invertible)

Let $[A]$ be an $n \times n$ diagonalizable matrix with a null eigenvalue. It may be written as

$$[A] = [S][\Lambda][S]^{-1}$$

Well,

$$\det([A]) = \det([S])\det([\Lambda])\det([S^{-1}]) \Rightarrow$$

(The determinant of the product is the product of the determinants)

$$\det([S]^{-1}) = \frac{1}{\det([S])} \text{ since}$$

$$\det([S][S]^{-1}) = \underbrace{\det([I])}_{=1} \Rightarrow$$

$$\Rightarrow \det([S])\det([S]^{-1}) = 1$$

$$\det([A]) = \prod_{i=1}^{n} \lambda_i \qquad (87)$$

$$\left(\det([\Lambda]) = \prod_{i=1}^{n} \lambda_i\right) \quad (\Rightarrow)$$

One may conclude from (87) that if there is an eigenvalue λ_i that is null the determinant of the matrix would also be null and the matrix would not be invertible.

In the case of the covariance matrix $[\Sigma]$, which is positive semi-definite, there could be a null eigenvalue and make $[\Sigma]$ to not be invertible!

However, in the model (79), $[\Sigma]^{-1}$ appears and one already makes the assumption that $[\Sigma]$ is invertible!

Note: (The transpose of the inverse is the inverse of the transpose)

One may show that the transpose of the inverse is the inverse of the transpose:

$$[A]^{-1,T} = \left(\frac{1}{\det([A])}\, Cof^{T}([A])\right)^{T} = \frac{1}{\det([A])}\, Cof([A]) =$$

$Cof([A]) = Cof^{T}([A]^{T}) \rightarrow = \frac{1}{\det([A]^{T})}\, Cof^{T}([A]^{T}) \quad (\leftarrow)$

$\det([A]) = \det([A]^{T})$

$$\boxed{[A]^{-1,T} = [A]^{T,-1}} \qquad (88)$$

(68)

Note: (The null space of an invertible matrix is zero)

One may show that an invertible matrix $[A]$ has solely zero $\{0\}$ as null space:

$$[A]\{x\} = \{0\} \Leftrightarrow$$

$$[A]^{-1}[A]\{x\} = [A]^{-1}\{0\} \Leftrightarrow$$

$$\Leftrightarrow \{x\} = \{0\}$$

Note: (The derivative of the determinant of an invertible matrix)

Let $[A(t)]$ be an invertible $n \times n$ matrix. Well,

$$\frac{d}{dt}(\det([A])) = \lim_{\Delta t \to 0} \frac{\det([A(t+\Delta t)]) - \det([A(t)])}{\Delta t} =$$

$$= \det([A(t)]) \lim_{\Delta t \to 0} \frac{\frac{\det([A(t+\Delta t)])}{\det([A(t)])} - 1}{\Delta t} =$$

$$= \det([A(t)]) \lim_{\Delta t \to 0} \frac{\det([A(t)]^{-1})\det([A(t+\Delta t)]) - 1}{\Delta t} =$$

$\left(\frac{1}{\det([A])} = \det([A]^{-1})\right)$

$$= \det([A(t)]) \lim_{\Delta t \to 0} \frac{\det([A(t)]^{-1}[A(t+\Delta t)]) - 1}{\Delta t} =$$

$\left(\det([A][B]) = \det([A]).\det([B])\right)$

$$= \det([A(t)]) \lim_{\Delta t \to 0} \frac{\det\left([A(t)]^{-1}.\left([A(t)] + \frac{d[A]}{dt}(t)\,\Delta t + [I]\,O(\Delta t^2)\right)\right) - 1}{\Delta t} \Rightarrow$$

(Taylor series expansion)

(69)

$\Rightarrow$ $\frac{d}{dt}\left(\det([A])\right) = \det([A(t)]) \cdot \lim_{\Delta t \to 0} \frac{\det\left([I] + \Delta t [A(t)]^{-1} \frac{d[A]}{dt}(t)\right) - 1}{\Delta t}$ $(=)$

$\det([I] + \Delta t [B]) = \begin{vmatrix} 1+\Delta t B_{11} & \Delta t B_{12} & \cdots \\ \Delta t B_{21} & 1+\Delta t B_{22} & \\ \vdots & & \ddots \end{vmatrix} =$

(Computing determinant using the forward and backward digonals)

$= \underbrace{(1+\Delta t B_{11})(1+\Delta t B_{22}) \cdots (1+\Delta t B_{mm})}_{= 1 + \Delta t \sum_{i=1}^{m} B_{ii} + O(\Delta t^2)} +$

$+ \underbrace{\Delta t B_{12}\, \Delta t B_{23} \cdots \Delta t B_{m-1,m}\, \Delta t B_{m1}}_{= O(\Delta t^2)}$

$+ \underbrace{\cdots}_{= O(\Delta t^2)}$

$+ \underbrace{\Delta t B_{1m} \cdot \Delta t B_{21}\, \Delta t B_{32} \cdots \Delta t B_{m,m-1}}_{= O(\Delta t^2)}$

$+ \underbrace{\Delta t B_{m1}\, \Delta t B_{m-1,2} \cdots \Delta t B_{1m}}_{= O(\Delta t^2)}$

$- \underbrace{\Delta t B_{m-1,1}\, \Delta t B_{m-2,2} \cdots \Delta t B_{1,m-1}\, \Delta t B_{mm}}_{= O(\Delta t^2)}$

$- \underbrace{\Delta t B_{11} \cdot \Delta t B_{m,2} \cdots \Delta t B_{2,m}}_{= O(\Delta t^2)}$

$= 1 + \Delta t \cdot \mathrm{trace}([B]) + O(\Delta t^2)$

$\Rightarrow$ $\frac{d}{dt}\left(\det([A])\right) = \det([A]) \lim_{\Delta t \to 0} \frac{1 + \Delta t\, \mathrm{trace}\left([A]^{-1} \frac{d[A]}{dt}\right) - 1}{\Delta t}$ $(=)$ $\leftarrow$

$\leftarrow$ $(=)$

$$\boxed{\frac{d}{dt}\left(\det[A]\right) = \det([A])\, \mathrm{trace}\left([A]^{-1} \frac{d[A]}{dt}\right)} \quad (89)$$

Let one now consider the case t is an entry of $[A]$, that is $t = A_{ij}$. In the case, (89) is transformed into

$$\frac{d}{dA_{ij}}\left(\det[A]\right) = \det([A])\, \mathrm{trace}\left([A]^{-1} \frac{d[A]}{dA_{ij}}\right) =$$

$\left([A] = \sum_{kl} A_{kl} \{e_k\}\{e_l\}^T\right)$

$$= \det([A])\, \mathrm{trace}\left([A]^{-1} \frac{d}{dA_{ij}} \sum_{kl} A_{kl} \{e_k\}\{e_l\}^T\right) =$$

$$= \det([A])\, \mathrm{trace}\left([A]^{-1} \sum_{k,l} \delta_{ik}\delta_{jl} \{e_k\}\{e_l\}^T\right) =$$

$$= \det([A])\, \mathrm{trace}\left([A]^{-1} \{e_i\}\{e_j\}^T\right) =$$

$$= \det([A])\, \mathrm{trace}\left(\sum_k [A]^{-1}_{ki} \{e_k\}\{e_j\}^T\right) =$$

$$= \det([A]) \left(\sum_k [A]^{-1}_{ki}\, \underbrace{\mathrm{trace}\left(\{e_k\}\{e_j\}^T\right)}_{= \delta_{kj}}\right) =$$

$$= \det([A]) \left(\sum_k [A]^{-1}_{ki}\, \delta_{kj}\right) = \det([A])\, [A]^{-1}_{ji} \quad (\Rightarrow$$

$(\Rightarrow$ $\boxed{\frac{d}{dA_{ij}}\left(\det[A]\right) = \det([A])\, [A]^{-1,T}_{ij}}$ (90)

Note: (Derivative of inverse of matrix)

Let one consider an invertible matrix $[A(t)]$. Well,

$$[A(t)][A(t)]^{-1} = [I] \Leftrightarrow \frac{d}{dt}\left([A(t)][A(t)]^{-1}\right) = [0] \Leftrightarrow$$

$$\Leftrightarrow \frac{d}{dt}\left(\left(\sum_{ij}[A]_{ij}\,\{e_i\}\{e_j\}^T\right)\left(\sum_{kl}[A]^{-1}_{kl}\,\{e_k\}\{e_l\}^T\right)\right) = [0] \Leftrightarrow$$

$$\Leftrightarrow \left(\left(\sum_{ij}\frac{d[A]_{ij}}{dt}\{e_i\}\{e_j\}^T\right)\left(\sum_{kl}[A]^{-1}_{kl}\{e_k\}\{e_l\}^T\right)+\right.$$

$$\left.+\left(\sum_{ij}[A]_{ij}\{e_i\}\{e_j\}^T\right)\left(\sum_{kl}\frac{d[A]^{-1}_{kl}}{dt}\{e_k\}\{e_l\}^T\right)\right) = [0] \Leftrightarrow$$

$$\Leftrightarrow \frac{d[A]}{dt}[A]^{-1} + [A]\frac{d[A]^{-1}}{dt} = [0] \Leftrightarrow [A]\frac{d[A]^{-1}}{dt} = -\frac{d[A]}{dt}[A]^{-1} \Leftrightarrow$$

$$\Leftrightarrow \boxed{\frac{d[A]^{-1}}{dt} = -[A]^{-1}\frac{d[A]}{dt}[A]^{-1}} \tag{91}$$

For the case of $t = A_{ij}$, (91) is transformed into

$$\frac{d[A]^{-1}}{dA_{ij}} = -[A]^{-1}\frac{d[A]}{dA_{ij}}[A]^{-1} = -[A]^{-1}\left(\sum_{kl}\frac{dA_{kl}}{dA_{ij}}\{e_k\}\{e_l\}^T\right)[A]^{-1} =$$

$$= -[A]^{-1}\left(\sum_{kl}\delta_{ki}\delta_{lj}\{e_k\}\{e_l\}^T\right)[A]^{-1} = -[A]^{-1}\{e_i\}\{e_j\}^T[A]^{-1} \in$$

$$\Rightarrow \boxed{\frac{d[A]^{-1}}{dA_{ij}} = -\underbrace{\left(\sum_k [A]^{-1}_{ki}\{e_k\}\right)}_{=\,i\text{-th column of } [A]^{-1}}\cdot\underbrace{\left(\sum_l \{e_l\}^T[A]^{-1}_{jl}\right)}_{=\,j\text{-th row of } [A]^{-1}}} \tag{92}$$

Let one further develop (83):

$$\frac{1}{2}\left([\Sigma]^{-1,T} + [\Sigma]^{-1}\right)\left(\left(\sum_{i=1}^{K}\{x_i\}\right) - K\{\mu^*\}\right) = \{0\} \Leftrightarrow \overset{(82)}{\underset{(85)}{\Leftarrow}}$$

$$\Leftrightarrow [\Sigma]^{-1}\left(\left(\sum_{i=1}^{K}\{x_i\}\right) - K\{\mu^*\}\right) = \{0\} \Leftrightarrow$$

(Since $[\Sigma]^{-1}$ is an invertible matrix its null space corresponds to $\{0\}$)

$$\Leftrightarrow \boxed{\{\mu^*\} = \frac{1}{K}\sum_{i=1}^{K}\{x_i\}} \tag{93}$$

(Optimum mean vector according to the Maximum Likelihood principle)

Note that

$$\frac{\partial^2 \mathcal{L}}{\partial(\mu^{(j)})^2}\Big|_{\mu=\mu^*} \overset{(82)}{=} -\left(K\{e_j\}^T[\Sigma]^{-1}\frac{\partial\{\mu\}}{\partial\mu^{(j)}}\right)\Big|_{\mu=\mu^*} = -\left(K\{e_j\}^T[\Sigma]^{-1}\frac{\partial}{\partial\mu^{(j)}}\left(\sum_{i=1}^{K}\mu^{(i)}\{e_i\}\right)\right)\Big|_{\mu=\mu^*} =$$

$$= -K\{e_j\}^T[\Sigma]^{-1}\sum_{i=1}^{K}\delta_{ij}\{e_i\} = -K\{e_j\}^T[\Sigma]^{-1}\{e_j\} \in$$

($\{e_j\}^T[\Sigma]^{-1}\{e_j\} > 0$ since $[\Sigma]$ is positive definite ($[\Sigma]$ is in general positive semi-definite but since $[\Sigma]$ is invertible, it is necessarily positive definite) and an inverse matrix of a positive definite one is also positive definite.)

(c) $$\left| \frac{\partial^2 \mathcal{L}}{\partial (\mu^{(j)})^2} \right|_{\mu=\mu^*} < 0 \quad \forall\, j = 1, \dots, m \tag{94}$$

From (94) one concludes that μ^* given by (93) maximizes the likelihood function $\mathcal{L}$.

Note: (The inverse of a positive definite matrix is also positive definite)

Let $[A]$ be a positive definite matrix. Since it is positive definite, it is also invertible (as it has no null eigenvalue). One has

$$\{N\}^T [A]^{-1} \{N\} = \{\mu\}^T [A]^T [A]^{-1} [A] \{\mu\} = \{\mu\}^T [A]^T \{\mu\} > 0 \quad \square$$

(Let $\{N\} = [A]\{\mu\}$)

since the transpose of a positive definite matrix is also positive definite.

Note: (The transpose of a positive definite matrix is also positive definite)

Let $[A]$ be a positive definite matrix. One has

$$\{N\}^T [A]^T \{N\} = \left(\{N\}^T [A] \{N\}\right)^T = \{N\}^T [A] \{N\} > 0 \quad \square$$

($\{N\}^T [A] \{N\}$ is a scalar)

Now,

$$\frac{\partial \mathcal{L}}{\partial \Sigma_{jl}} \overset{(81)}{=} -\frac{k}{2} \frac{\frac{\partial}{\partial \Sigma_{jl}} |[\Sigma]|}{|[\Sigma]|} - \frac{1}{2} \sum_{i=1}^{k} \left(\{x_i\} - \{\mu\}\right)^T \frac{\partial [\Sigma]^{-1}}{\partial \Sigma_{jl}} \left(\{x_i\} - \{\mu\}\right) =$$

$$= -\frac{k}{2} \frac{|[\Sigma]| \, [\Sigma]^{-1,T}_{jl}}{|[\Sigma]|} -$$

(90), (92)

$$+ \frac{1}{2} \sum_{i=1}^{k} \left(\{x_i\} - \{\mu\}\right)^T \left(\sum_m [\Sigma]^{-1}_{mj} \{e_m\}\right) \cdot \left(\sum_n \{e_n\}^T [\Sigma]^{-1}_{ln}\right) \left(\{x_i\} - \{\mu\}\right) =$$

$$= -\frac{k}{2} [\Sigma]^{-1}_{jl} +$$

$$\left([\Sigma]^{-1T} = [\Sigma]^{T-1} = [\Sigma]^{-1} \right)$$

$$+ \frac{1}{2} \sum_{i=1}^{k} \sum_{m} \sum_{s} \left(x_i^{(s)} - \mu^{(s)}\right) \underbrace{\{e_s\}^T \{e_m\}}_{=\delta_{sm}} [\Sigma]^{-1}_{mj} \sum_{n} \sum_{t} [\Sigma]^{-1}_{ln} \left(x_i^{(t)} - \mu^{(t)}\right) \underbrace{\{e_n\}^T \{e_t\}}_{=\delta_{nt}} =$$

$$= -\frac{k}{2} [\Sigma]^{-1}_{jl} + \frac{1}{2} \sum_{i=1}^{k} \left(\sum_m \left(x_i^{(m)} - \mu^{(m)}\right) [\Sigma]^{-1}_{mj}\right) \left(\sum_n [\Sigma]^{-1}_{ln} \left(x_i^{(n)} - \mu^{(n)}\right)\right)$$

$$= -\frac{k}{2} [\Sigma]^{-1}_{jl} + \frac{1}{2} \sum_{i=1}^{k} \left(\sum_m [\Sigma]^{-1}_{jm} \left(x_i^{(m)} - \mu^{(m)}\right)\right) \left(\sum_n \left(x_i^{(n)} - \mu^{(n)}\right) [\Sigma]^{-1}_{nl}\right) =$$

($[\Sigma]^{-1}$ is symmetric, and, therefore $[\Sigma]^{-1}_{mj} = [\Sigma]^{-1}_{jm}$ and $[\Sigma]^{-1}_{ln} = [\Sigma]^{-1}_{nl}$)

$$= -\frac{k}{2} \{e_j\}^T [\Sigma]^{-1} \{e_l\} + \frac{1}{2} \{e_j\}^T [\Sigma]^{-1} \left[\sum_{i=1}^{k} \left(\{x_i\} - \{\mu\}\right) \left(\{x_i\} - \{\mu\}\right)^T\right] [\Sigma]^{-1} \{e_l\} (=$$

$$\Rightarrow \boxed{\frac{\partial \mathcal{L}}{\partial \Sigma_{jl}} = \{e_j\}^T \left(-\frac{k}{2}[\Sigma]^{-1} + \frac{1}{2}[\Sigma]^{-1}\left(\sum_{i=1}^{k}(\{x_i\}-\{\mu\})(\{x_i\}-\{\mu\})^T\right)[\Sigma]^{-1}\right)\{e_l\}} \tag{95}$$

$$\frac{\partial \mathcal{L}}{\partial \Sigma_{jl}}\Big|_{\substack{\mu=\mu^* \\ \Sigma_{jl}=\Sigma^*_{jl}}} = 0 \quad \forall j,l \Rightarrow$$

$$\Rightarrow -\frac{k}{2}[\Sigma^*]^{-1} + \frac{1}{2}[\Sigma^*]^{-1}\left(\sum_{i=1}^{k}(\{x_i\}-\{\mu^*\})(\{x_i\}-\{\mu^*\})^T\right)[\Sigma^*]^{-1} = [0] \Rightarrow$$

$$\Rightarrow -k[\Sigma^*] + \left(\sum_{i=1}^{k}(\{x_i\}-\{\mu^*\})(\{x_i\}-\{\mu^*\})^T\right) = [0] \Rightarrow$$

$$\Rightarrow \boxed{[\Sigma^*] = \frac{1}{k}\sum_{i=1}^{k}(\{x_i\}-\{\mu^*\})(\{x_i\}-\{\mu^*\})^T} \tag{96}$$

(Optimum covariance matrix according to the Maximum Likelihood principle)

Not Hot

$$\frac{\partial^2 \mathcal{L}}{\partial \Sigma_{jl}^2}\Big|_{\substack{\mu=\mu^* \\ \Sigma_{jl}=\Sigma^*_{jl}}} \overset{(92)}{=} \{e_j\}^T\left(\frac{k}{2}\left(\sum_m [\Sigma^*]^{-1}_{mj}\{e_m\}\right)\cdot\left(\sum_n \{e_n\}^T[\Sigma^*]^{-1}_{ln}\right) + \right.$$

$$\left. - \frac{1}{2}\left(\sum_m [\Sigma^*]^{-1}_{mj}\{e_m\}\right)\cdot\left(\sum_n \{e_n\}^T[\Sigma^*]^{-1}_{ln}\right)\left(\sum_{i=1}^{k}(\{x_i\}-\{\mu^*\})(\{x_i\}-\{\mu^*\})^T\right)[\Sigma^*]^{-1}\right)$$

$$-\frac{1}{2}[\Sigma^*]^{-1}\left(\sum_{i=1}^{k}(\{x_i\}-\{\mu^*\})(\{x_i\}-\{\mu^*\})^T\right)\left(\sum_m [\Sigma^*]^{-1}_{mj}\{e_m\}\right)\cdot\left(\sum_n \{e_n\}^T[\Sigma^*]^{-1}_{ln}\right)\{e_l\} =$$

$$= -\frac{k}{2}\{e_j\}^T[\Sigma^*]^{-1}\{e_j\}\{e_l\}^T[\Sigma^*]^{-1}\{e_l\}$$

$$-\frac{1}{2}\{e_j\}^T[\Sigma^*]^{-1}\{e_j\}\{e_l\}^T[\Sigma^*]^{-1}\underbrace{\left(\sum_{i=1}^{k}(\{x_i\}-\{\mu^*\})(\{x_i\}-\{\mu^*\})^T\right)}_{=k[\Sigma^*]}[\Sigma^*]^{-1}\{e_l\}$$

$$-\frac{1}{2}\{e_j\}^T[\Sigma^*]^{-1}\underbrace{\left(\sum_{i=1}^{k}(\{x_i\}-\{\mu^*\})(\{x_i\}-\{\mu^*\})^T\right)}_{=k[\Sigma^*]}[\Sigma^*]^{-1}\{e_j\}\{e_l\}^T[\Sigma^*]^{-1}\{e_l\} =$$

$$= \frac{k}{2}\{e_j\}^T[\Sigma^*]^{-1}\{e_j\}\{e_l\}^T[\Sigma^*]^{-1}\{e_l\}$$

$$-\frac{k}{2}\{e_j\}^T[\Sigma^*]^{-1}\{e_j\}\{e_l\}^T[\Sigma^*]^{-1}\{e_l\}$$

$$-\frac{k}{2}\{e_j\}^T[\Sigma^*]^{-1}\{e_j\}\{e_l\}^T[\Sigma^*]^{-1}\{e_l\} \Rightarrow$$

$$\Rightarrow \frac{\partial^2 \mathcal{L}}{\partial \Sigma_{jl}^2}\Big|_{\substack{\mu=\mu^* \\ \Sigma_{jl}=\Sigma^*_{jl}}} = -\frac{k}{2}\underbrace{\{e_j\}^T[\Sigma^*]^{-1}\{e_j\}}_{>0}\{e_l\}^T[\Sigma^*]^{-1}\{e_l\} \Rightarrow$$

>0 since $[\Sigma^*]^{-1}$ is positive definite

$$\Rightarrow \boxed{\frac{\partial^2 \mathcal{L}}{\partial \Sigma_{jl}^2}\Big|_{\substack{\mu=\mu^* \\ \Sigma_{jl}=\Sigma^*_{jl}}} < 0 \quad \forall\, j,l = 1,\dots,m} \tag{97}$$

Result (97) shows that $[\Sigma^*]$ given by (96) maximizes the likelihood function $\mathcal{L}$.

Note: (Estimators' biases)

The estimator $[\Sigma^*]$ given by (96) is actually biased, that is

$$\mathbb{E}\left([\Sigma] - [\Sigma^*]\right) \neq [0]$$

where $[\Sigma]$ is the true covariance matrix.

Let

$$\boxed{\{\mu\} := \mathbb{E}(\{X\}) = \mathbb{E}(\{x_i\})} \tag{98}$$

(True mean vector)

be the true mean vector. One may show that the estimate mean vector is unbiased:

$$\mathbb{E}(\{\mu^*\} - \{\mu\}) \overset{(93)}{=} \mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{k}\{x_i\} - \{\mu\}\right) \overset{\left(\mathbb{E}\text{ is a linear operator}\right)}{=}$$

$$= \frac{1}{k}\sum_{i=1}^{k}\underbrace{\mathbb{E}(\{x_i\})}_{=\{\mu\}\text{ from (98)}} - \{\mu\}\underbrace{\mathbb{E}(1)}_{=1} = \frac{1}{k}\left(\sum_{i=1}^{k}\{\mu\}\right) - \{\mu\} = \{\mu\} - \{\mu\} \quad (\Rightarrow)$$

$$\Rightarrow \boxed{\mathbb{E}(\{\mu^*\} - \{\mu\}) = \{0\}} \tag{99}$$

(Mean vector estimator's bias) →

← The true covariance matrix is defined as $[\Sigma]$ so that

$$\Sigma_{jl} = \mathbb{E}\left((X^{(j)} - \mu^{(j)})(X^{(l)} - \mu^{(l)})\right) \Leftrightarrow$$

$$\Leftrightarrow [\Sigma] = \sum_{jl}\sum_{jl}\{e_j\}\{e_l\}^T = \mathbb{E}\left(\sum_j \underbrace{(X^{(j)} - \mu^{(j)})\{e_j\}}_{=\{X\} - \{\mu\}}\sum_l \underbrace{(X^{(l)} - \mu^{(l)})\{e_l\}^T}_{=(\{X\} - \{\mu\})^T}\right) \Leftrightarrow$$

$\left(\mathbb{E}\text{ is a linear operator}\right)$

$$\Leftrightarrow \boxed{[\Sigma] = \mathbb{E}\left((\{X\} - \{\mu\})(\{X\} - \{\mu\})^T\right) = \mathbb{E}\left((\{x_i\} - \{\mu\})\cdot(\{x_i\} - \{\mu\})^T\right)} \tag{100}$$

(True covariance matrix)

One may show that the estimate covariance matrix is biased:

$\left(\mathbb{E}\text{ is a linear operator}\right)$

$$\mathbb{E}\left([\Sigma^*] - [\Sigma]\right) \overset{(96)}{=} \mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{k}(\{x_i\} - \{\mu^*\})(\{x_i\} - \{\mu^*\})^T\right) - [\Sigma]\underbrace{\mathbb{E}(1)}_{=1} =$$

$$= \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}\left(\sum_j (x_i^{(j)} - \mu^{*(j)})\{e_j\}\sum_l (x_i^{(l)} - \mu^{*(l)})\{e_l\}^T\right) - [\Sigma] =$$

$\left(\mathbb{E}\text{ is a linear operator}\right)$

$$= \frac{1}{k}\sum_{i=1}^{k}\sum_{jl}\{e_j\}\{e_l\}^T\,\mathbb{E}\left((x_i^{(j)} - \mu^{*(j)})(x_i^{(l)} - \mu^{*(l)})\right) - [\Sigma] =$$

$$= \frac{1}{k}\sum_{i=1}^{k}\sum_{jl}\{e_j\}\{e_l\}^T\,\mathbb{E}\left(x_i^{(j)}x_i^{(l)} - x_i^{(j)}\mu^{*(l)} - x_i^{(l)}\mu^{*(j)} + \mu^{*(j)}\mu^{*(l)}\right) - [\Sigma] \quad (\Rightarrow)$$

→

$$(100): [\Sigma] = \mathbb{E}\left(\left(\{x_i\} - \{\mu\}\right)\cdot\left(\{x_i\} - \{\mu\}\right)^T\right) \;\forall i = 1,\dots,k \;(\Leftrightarrow$$

$$\Leftrightarrow \frac{1}{k}\underbrace{\sum_{i=1}^{k}[\Sigma]}_{=k[\Sigma]} = \mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{k}\left(\{x_i\} - \{\mu\}\right)\cdot\left(\{x_i\} - \{\mu\}\right)^T\right) (\Leftrightarrow$$

$$\Leftrightarrow [\Sigma] = \mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{k}\left(\{x_i\} - \{\mu\}\right)\cdot\left(\{x_i\} - \{\mu\}\right)^T\right)$$

$$= \frac{1}{k}\sum_{i=1}^{k}\sum_{jl}\{e_j\}\{e_l\}^T\,\mathbb{E}\left(x_i^{(j)}x_i^{(l)} - x_i^{(j)}\mu^{(l)} - x_i^{(l)}\mu^{(j)} + \mu^{(j)}\mu^{(l)}\right)$$

$$(\Leftrightarrow \mathbb{E}\left([\Sigma^*] - [\Sigma]\right) = \frac{1}{k}\sum_{i=1}^{k}\sum_{jl}\{e_j\}\{e_l\}^T\,\mathbb{E}\left(\mu^{*(j)}\mu^{*(l)} - x_i^{(j)}(\mu^{*(l)} - \mu^{(l)})\right.$$

$$\left. - x_i^{(l)}(\mu^{*(j)} - \mu^{(j)}) - \mu^{(j)}\mu^{(l)}\right) =$$

$$= \sum_{jl}\{e_j\}\{e_l\}^T\,\mathbb{E}\left(\underbrace{\left(\frac{1}{k}\sum_{i=1}^{k}1\right)}_{=1}\mu^{*(j)}\mu^{*(l)} - \underbrace{\left(\frac{1}{k}\sum_{i=1}^{k}x_i^{(j)}\right)}_{=\mu^{*(j)}}(\mu^{*(l)} - \mu^{(l)})\right.$$

$$- \underbrace{\left(\frac{1}{k}\sum_{i=1}^{k}x_i^{(l)}\right)}_{=\mu^{*(l)}}(\mu^{*(j)} - \mu^{(j)})$$

$$\left. - \underbrace{\left(\frac{1}{k}\sum_{i=1}^{k}1\right)}_{=1}\mu^{(j)}\mu^{(l)}\right) (=$$

$$(\Leftrightarrow \mathbb{E}\left([\Sigma^*] - [\Sigma]\right) = \sum_{jl}\{e_j\}\{e_l\}^T\,\mathbb{E}\left(\mu^{*(j)}\mu^{*(l)} - \mu^{*(j)}\mu^{*(l)} + \mu^{*(j)}\mu^{(l)}\right.$$

$$\left. - \mu^{*(l)}\mu^{*(j)} + \mu^{*(l)}\mu^{(j)} - \mu^{(j)}\mu^{(l)}\right)$$

$$= -\sum_{jl}\{e_j\}\{e_l\}^T\,\mathbb{E}\left(\mu^{*(l)}\mu^{*(j)} - \mu^{*(j)}\mu^{(l)} - \mu^{*(l)}\mu^{(j)} + \mu^{(j)}\mu^{(l)}\right)$$

$$= -\mathbb{E}\left(\left(\sum_{j}(\mu^{*(j)} - \mu^{(j)})\{e_j\}\right)\left(\sum_{l}(\mu^{*(l)} - \mu^{(l)})\{e_l\}^T\right)\right) =$$

$$= -\mathbb{E}\left(\left(\{\mu^*\} - \underbrace{\{\mu\}}_{=\mathbb{E}(\{\mu^*\})}\right)\left(\{\mu^*\} - \underbrace{\{\mu\}}_{=\mathbb{E}(\{\mu^*\})}\right)^T\right) =$$

$$= -[\Sigma_{\{\mu^*\}}] \;(=$$

(Covariance of $\{\mu^*\}$ vector) (Bienaymé's identity)

$$(\Leftrightarrow \boxed{\mathbb{E}\left([\Sigma^*] - [\Sigma]\right) = -\frac{[\Sigma]}{k}} \quad (101)$$

(Covariance estimator's bias)

From (101), one may define an unbiased estimator for the covariance:

$$(101): \mathbb{E}\left([\Sigma^*] - \frac{k-1}{k}[\Sigma]\right) = 0 \;(\Leftrightarrow\; \mathbb{E}\left(\frac{k}{k-1}[\Sigma^*] - \underbrace{[\Sigma]}_{=[S]}\right) = 0$$

$$\boxed{[S] = \frac{1}{k-1}\sum_{i=1}^{k}\left(\{x_i\} - \{\mu^*\}\right)\left(\{x_i\} - \{\mu^*\}\right)^T} \quad (102)$$

(Unbiased estimator for covariance matrix)

Note: (Covariance matrix of scaled vector variable, $[\Sigma_{a\{X\}}]$)

The covariance matrix of the vector variable $\{X\}$, scaled by the constant a corresponds to

$$[\Sigma_{a\{X\}}] = \mathbb{E}\left(\left(a\{X\} - \mathbb{E}(a\{X\})\right)\left(a\{X\} - \mathbb{E}(a\{X\})\right)^T\right) =$$

$$= a^2\, \mathbb{E}\left(\left(\{X\} - \mathbb{E}(\{X\})\right)\left(\{X\} - \mathbb{E}(\{X\})\right)^T\right) =$$

$$= [\Sigma_{\{X\}}]$$

(The expected value operator is linear)

$$\Rightarrow \boxed{[\Sigma_{a\{X\}}] = a^2 [\Sigma_{\{X\}}]} \quad (103)$$

Note: (Bienaymé's identity)

The Bienaymé's identity is such that

$$\boxed{[\Sigma_{\{\bar{x}\}}] = \frac{1}{k}[\Sigma_{\{X\}}]}, \quad (104)$$

where $\{\bar{x}\} = \frac{1}{K}\sum_{i=1}^{K} \{x_i\}$, and $\{x_i\}$ is a random value of the variable $\{X\}$.

← Let us show the identity (104). Well,

$$[\Sigma_{\{\bar{x}\}}] = \mathbb{E}\left(\left(\frac{1}{k}\sum_{i=1}^{K}\{x_i\} - \underbrace{\mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{K}\{x_i\}\right)}_{=\{\mu\},\ \text{from (99)}}\right)\cdot\left(\frac{1}{k}\sum_{j=1}^{K}\{x_j\} - \underbrace{\mathbb{E}\left(\frac{1}{k}\sum_{j=1}^{K}\{x_j\}\right)}_{=\{\mu\}}\right)^T\right)$$

$$= \frac{1}{k^2}\mathbb{E}\left(\left(\left(\sum_{i=1}^{K}\{x_i\}\right) - k\{\mu\}\right)\left(\left(\sum_{j=1}^{K}\{x_j\}\right) - k\{\mu\}\right)^T\right) =$$

$$= \frac{1}{k^2}\,\mathbb{E}\left(\left(\sum_{i=1}^{K}\left(\{x_i\} - \{\mu\}\right)\right)\left(\sum_{j=1}^{K}\left(\{x_j\} - \{\mu\}\right)\right)^T\right) =$$

$$= \frac{1}{k^2}\sum_{i=1}^{K}\sum_{j=1}^{K}\underbrace{\mathbb{E}\left(\left(\{x_i\} - \{\mu\}\right)\left(\{x_j\} - \{\mu\}\right)^T\right)}_{\text{= Covariance matrix between vectors } \{x_i\} \text{ and } \{x_j\}} =$$

(the elements of the sample are independent and, therefore, $\mathbb{E}\left(\left(\{x_i\} - \{\mu\}\right)\left(\{x_j\} - \{\mu\}\right)^T\right) = \underbrace{\mathbb{E}\left(\{x_i\} - \{\mu\}\right)}_{=\{0\}}\underbrace{\mathbb{E}\left(\{x_j\} - \{\mu\}\right)^T}_{=\{0\}^T} = [0]$, if $i \neq j$.)

$$= \frac{1}{k^2}\sum_{i=1}^{K}\mathbb{E}\left(\left(\{x_i\} - \{\mu\}\right)\left(\{x_i\} - \{\mu\}\right)^T\right) =$$

$$= \frac{1}{k^2}\sum_{i=1}^{K}\underbrace{\mathbb{E}\left(\left(\{X\} - \{\mu\}\right)\left(\{X\} - \{\mu\}\right)^T\right)}_{=[\Sigma_{\{X\}}]} =$$

(the elements of the sample are identically distributed)

$$= \frac{1}{K}[\Sigma_{\{X\}}] \quad \square$$

Incremental Bayes Classifier

Let one consider an experiment in which a sample of k examples which are conditionally independent and identically distributed under the hypothesis H: $\mathcal{S}_k = \{x_1, \dots, x_k\}$. Here, x_i is a value of the variable X. Let one also consider that the conditional probability $P(H | x_1, \dots, x_k)$ was previously obtained from the sample $\mathcal{S}_k$. It would be interesting to determine this probability when additionally considering a new example, x_{k+1}. Well,

$$P(H | x_1, \dots, x_k, x_{k+1}) \overset{(52)}{=} \frac{P(H, x_1, \dots x_k, x_{k+1})}{P(x_1, \dots, x_k, x_{k+1})} \overset{(52)}{=}$$

$$= \frac{P(x_{k+1} | x_1, \dots, x_k, H)\, P(x_1 \dots x_k, H)}{P(x_{k+1} | x_1, \dots, x_k)\, P(x_1, \dots, x_k)} \overset{(52)}{=}$$

$$= \frac{P(x_{k+1} | x_1, \dots, x_k, H)\, P(H | x_1, \dots, x_k) P(x_1, \dots, x_k)}{P(x_{k+1} | x_1, \dots x_k)\, P(x_1, \dots, x_k)} =$$

$$= \frac{P(x_{k+1} | H)\, P(H | x_1, \dots, x_k)}{P(x_{k+1} | x_1, \dots, x_k)} \quad (84)$$

(Since the examples are conditionally independent one has from (66): $P(x_{k+1} | x_1, \dots, x_k, H) = P(x_{k+1} | H)$)

$$P(x_{k+1} | x_1, \dots, x_k) \overset{(52)}{=} \frac{P(x_1, \dots, x_k, x_{k+1})}{P(x_1, \dots, x_k)} \overset{(47)}{=}$$

$$= \frac{\sum_i P(x_1, \dots, x_k, x_{k+1}, h_{(i)})}{P(x_1, \dots, x_k)} \overset{\text{(Here, } h_{(i)} \text{ is the } i\text{-th that } H \text{ may take)}}{=} \sum_i^K (52)$$

$$= \frac{\sum_i P(x_1, \dots, x_k, x_{k+1} | h_{(i)})\, P(h_{(i)})}{P(x_1, \dots, x_k)} =$$

$$= \frac{\sum_i P(x_{k+1} | h_{(i)}) P(x_1, \dots, x_k | h_{(i)})\, P(h_{(i)})}{P(x_1, \dots, x_k)} =$$

$$= \sum_i P(x_{k+1} | h_{(i)})\, P(h_{(i)} | x_1, \dots, x_k) \quad (56)$$

(Since the examples are conditionally independent one has from (67) that $P(x_1, \dots, x_k, x_{k+1} | h_{(i)}) = P(x_{k+1} | h_{(i)}) \cdot P(x_1, \dots, x_k | h_{(i)})$)

$$\boxed{P(h_{(i)} | x_1, \dots, x_k, x_{k+1}) = \frac{P(x_{k+1} | h_{(i)}) \cdot P(h_{(i)} | x_1, \dots, x_k)}{\left(\sum_j P(x_{k+1} | h_{(j)}) \cdot P(h_{(j)} | x_1, \dots, x_k) \right)}} \quad (105)$$

(Conditional probability of the hypothesis H being $h_{(i)}$ given a sample of $k+1$ examples which are conditionally independent and identically distributed under each hypothesis)

Bayes Classifier for 2-class Normal Distributions

Theorem: If $p(h_1) = p(h_2)$ and $p(\{x\}|h_1) \sim N(\{\mu_1\}, [\Sigma])$ and $p(\{x\}|h_2) \sim N(\{\mu_2\}, [\Sigma])$, the Bayes optimal decision surface (obtained from the Maximum A Posteriori principle) is a hyperplane $\{w\}^T(\{x\} - \{\mu\}) = 0$ where $\{\mu\} = \frac{\{\mu_1\} + \{\mu_2\}}{2}$ and $\{w\} = [\Sigma]^{-1}(\{\mu_1\} - \{\mu_2\})$.

Demonstration:

From (77) one has

$$\{h^*\} = \underset{\{h\}}{\arg\max}\, p(\{h\}|\{x\}) \overset{\text{(}\{h\} \text{ may take one of two discrete values)}}{=} \underset{\{h_1\},\{h_2\}}{\arg\max} \{p(\{h_1\}|\{x\}), p(\{h_2\}|\{x\})\}$$

The above statement means that h_1 would be chosen if $p(\{h_1\}|\{x\}) > p(\{h_2\}|\{x\})$ and h_2 if otherwise. The decision surface would then be defined as

$$S = \{\{x\} : p(\{h_1\}|\{x\}) - p(\{h_2\}|\{x\}) = 0\} \quad (106)$$

The condition (106) is equivalent to

$$\frac{p(\{h_1\}|\{x\})}{p(\{h_2\}|\{x\})} = 1 \Leftrightarrow \log\left(\frac{p(\{h_1\}|\{x\})}{p(\{h_2\}|\{x\})}\right) = 0 \Leftrightarrow \quad \text{(Bayes' theorem, (56))}$$

$$\Leftrightarrow \log\left(\frac{\frac{p(\{x\}|\{h_1\})p(\{h_1\})}{p(\{x\})}}{\frac{p(\{x\}|\{h_2\})p(\{h_2\})}{p(\{x\})}}\right) = 0 \Leftrightarrow \quad \text{(} p(\{h_1\}) = p(\{h_2\}) \text{, making the MAP principle be transformed into ML one)}$$

$$\Leftrightarrow \log\left(\frac{p(\{x\}|\{h_1\})}{p(\{x\}|\{h_2\})}\right) = 0 \Leftrightarrow$$

(86)

$\Leftarrow$

$$\Leftarrow \log\left(p(\{x\}|\{h_1\})\right) - \log\left(p(\{x\}|\{h_2\})\right) = 0 \Leftarrow \quad \text{(} p(\{x\}|\{h_1\}) \text{ and } p(\{x\}|\{h_2\}) \text{ correspond to Gaussian probability density functions, (79), with } \{\mu\} = \{\mu_1\} \text{ and } \{\mu\} = \{\mu_2\}, \text{ respectively.)}$$

$$\Leftarrow \log\left(\frac{e^{-\frac{1}{2}(\{x\}-\{\mu_1\})^T[\Sigma]^{-1}(\{x\}-\{\mu_1\})}}{\sqrt{(2\pi)^m|[\Sigma]|}}\right) - \log\left(\frac{e^{-\frac{1}{2}(\{x\}-\{\mu_2\})^T[\Sigma]^{-1}(\{x\}-\{\mu_2\})}}{\sqrt{(2\pi)^m|[\Sigma]|}}\right) = 0 \Leftarrow$$

$$\Leftarrow -\frac{1}{2}(\{x\}-\{\mu_1\})^T[\Sigma]^{-1}(\{x\}-\{\mu_1\}) + \frac{1}{2}(\{x\}-\{\mu_2\})^T[\Sigma]^{-1}(\{x\}-\{\mu_2\}) = 0 \Leftarrow$$

$$\Leftarrow -\frac{1}{2}\{x\}^T[\Sigma]^{-1}\{x\} + \frac{1}{2}\{x\}^T[\Sigma]^{-1}\{\mu_1\} + \frac{1}{2}\{\mu_1\}^T[\Sigma]^{-1}\{x\} - \frac{1}{2}\{\mu_1\}^T[\Sigma]^{-1}\{\mu_1\}$$

$$+\frac{1}{2}\{x\}^T[\Sigma]^{-1}\{x\} - \frac{1}{2}\{x\}^T[\Sigma]^{-1}\{\mu_2\} - \frac{1}{2}\{\mu_2\}^T[\Sigma]^{-1}\{x\} + \frac{1}{2}\{\mu_2\}^T[\Sigma]^{-1}\{\mu_2\} = 0 \Leftarrow$$

$$\Leftarrow \frac{1}{2}\{\mu_1\}^T[\Sigma]^{-1,T}\{x\} + \frac{1}{2}\{\mu_1\}^T[\Sigma]^{-1,T}\{x\} - \frac{1}{2}\{\mu_2\}^T[\Sigma]^{-1,T}\{x\} - \frac{1}{2}\{\mu_2\}^T[\Sigma]^{-1,T}\{x\} - \frac{1}{2}\{\mu_1\}^T[\Sigma]^{-1,T}\{\mu_1\}$$

$$+\frac{1}{2}\{\mu_2\}^T[\Sigma]^{-1,T}\{\mu_2\} = 0 \Leftarrow$$

($[\Sigma]^{-1,T} = [\Sigma]^{-1}$)

($\{x\}^T[\Sigma]^{-1}\{\mu_2\}$ is a scalar, and, therefore, $\{x\}^T[\Sigma]^{-1}\{\mu_1\} = (\{x\}^T[\Sigma]^{-1}\{\mu_1\})^T = \{\mu_2\}^T[\Sigma]^{-1,T}\{x\}$)

$$\Leftarrow (\{\mu_1\}^T - \{\mu_2\}^T)[\Sigma]^{-1,T}\{x\} - \{\mu_1\}^T[\Sigma]^{-1,T}\frac{1}{2}\{\mu_1\} + \{\mu_2\}^T[\Sigma]^{-1,T}\frac{1}{2}\{\mu_2\} = 0 \Leftarrow$$

$$\Leftarrow \left([\Sigma]^{-1}(\{\mu_1\}-\{\mu_2\})\right)^T\{x\} - (\{\mu_1\}^T - \{\mu_2\}^T)[\Sigma]^{-1,T}\cdot\frac{1}{2}\{\mu_1\} - (\{\mu_1\}^T - \{\mu_2\}^T)[\Sigma]^{-1,T}\cdot\frac{1}{2}\{\mu_2\} =$$

$$-\{\mu_2\}^T[\Sigma]^{-1,T}\cdot\frac{1}{2}\{\mu_1\} + \{\mu_1\}^T[\Sigma]^{-1,T}\cdot\frac{1}{2}\{\mu_2\} = 0 \quad \Leftarrow$$

(87)

$\Leftarrow$

$\Leftrightarrow \left([\Sigma]^{-1} (\vec{\mu}_1 - \vec{\mu}_2) \right)^T \vec{x} - \left([\Sigma]^{-1} (\vec{\mu}_1 - \vec{\mu}_2) \right)^T \left(\frac{\vec{\mu}_1 + \vec{\mu}_2}{2} \right) = 0 \Leftrightarrow$

$\left(\vec{\mu}_1^T [\Sigma]^{-1,T} \cdot \frac{1}{2} \vec{\mu}_2 = (\vec{\mu}_1^T [\Sigma]^{-1,T} \cdot \frac{1}{2} \vec{\mu}_2)^T = \right.$
$= \vec{\mu}_2^T \underbrace{[\Sigma]^{-1}}_{=[\Sigma]^{-1,T}} \cdot \frac{1}{2} \vec{\mu}_1 = \vec{\mu}_2^T [\Sigma]^{-1,T} \cdot \frac{1}{2} \vec{\mu}_1$
since $\vec{\mu}_1^T [\Sigma]^{-1,T} \cdot \frac{1}{2} \vec{\mu}_2$ is a scalar$\left.\right)$

$\Leftrightarrow \underbrace{\left([\Sigma]^{-1} (\vec{\mu}_1 - \vec{\mu}_2) \right)^T}_{=: \vec{w}^T} \left(\vec{x} - \underbrace{\frac{\vec{\mu}_1 + \vec{\mu}_2}{2}}_{=: \vec{\mu}} \right) = 0 \Leftrightarrow$

$\Leftrightarrow \vec{w}^T (\vec{x} - \vec{\mu}) = 0$ $\square$

II - Maximum Likelihood / Maximum Entropy Duality

The Maximum Likelihood principle and Empirical Distribution

Let's consider a variable $\{X\}$ that may take a value that pertains to the set $\mathcal{X} = \{\{a_1\}, \dots, \{a_n\}\}$. Also, let $P(\{a_i\}|\{\Theta\}) = P(\{X = \{a_i\}|\{\Theta\})$ be the probability of $\{X\}$ being $\{a_i\}$ given the set of parameters $\{\Theta\}$.

Now, consider an experience in which values of $\{X\}$ are successively taken, being these $\{x_1\}, \dots, \{x_m\}$, and that the experience is such that $\{x_1\}, \dots \{x_m\}$ are independent and identically distributed according to the probability $P(\{X\} = \{a_i\}|\{\Theta\})$. Finally let the occurrence frequency $f(a)$ be the number of occurrences of the value a in $\{x_1\}, \dots, \{x_m\}$:

$$\boxed{f(a) = |\{i : \{x_i\} = \{a\}\}|} \quad (107)$$

(Occurrence frequency)

The respective empirical distribution is defined as the estimate of the probability of the value a being taken in the experiment which is given by

$$\hat{P}(\{a_i\}) = \frac{f(\{a_i\})}{\sum_{\{x\}\in\mathcal{X}} f(\{x\})} \quad (=) \quad \boxed{\hat{P}(\{a_i\}) = \hat{P}_i(\{x\}=\{a_i\}) = \frac{f(\{a_i\})}{m}} \quad (108)$$

(Empirical distribution)

$$\left((107): \sum_{\{x\}\in\mathcal{X}} f(\{x\}) = \sum_{\{x\}\in\mathcal{X}} |\{i : \{x_i\}=\{x\}\}| = \left| \bigcup_{\{x\}\in\mathcal{X}} \{i : \{x_i\}=\{x\}\} \right| = |\{i : \{x_i\}\in\mathcal{X}\}| = m, \text{ number of examples} \right)$$

Well, the joint probability of the examples $\{x_1\}, \dots, \{x_m\}$ taking their values given the set of parameters $\{\theta\}$ is

$$P(\{x_1\},\dots,\{x_m\}|\{\theta\}) = \prod_{i=1}^{m} P(\{x_i\}|\{\theta\}) = \prod_{\{a\}\in\mathcal{X}} P(\{a\}|\{\theta\})^{f(\{a\})} \quad (=)$$

($\{x_1\},\dots,\{x_m\}$ are i.i.d.)

$$(=) \quad \boxed{P(\{x_1\},\dots,\{x_m\}|\{\theta\}) = \prod_{i=1}^{n} P_i^{f_i}}, \quad (109)$$

where $\boxed{P_i := P(\{a_i\}|\{\theta\})}$ (110) and $\boxed{f_i := f(\{a_i\})}$ (111).

The Maximum Likelihood principle, (61), applied to the current problem may be expressed as

$$\{\theta^*\} = \underset{\{\theta\}}{\operatorname{argmax}}\ P(\{x_1\},\dots,\{x_m\}|\{\theta\}) \quad (=) \quad (109)$$

$$(=) \quad \boxed{\{P^*\} = \underset{\{P\}\in Q}{\operatorname{argmax}} \prod_{i=1}^{n} P_i^{f_i}}, \quad (112)$$

where $\boxed{Q = \{\{q\}\in\mathbb{R}^n : q_i \geq 0,\ \sum_{i=1}^{n} q_i = 1\}}$ (113) corresponds to the set of n-dimensional probability vectors ("probability simplex"). By taking into account the monotonicity of the logarithmic function, equation (112) may be simplified to

$$\boxed{\begin{array}{l} \underset{\{P\}}{\operatorname{argmax}} \sum_{i=1}^{n} f_i \ln(P_i) \\ \text{subject to} \\ \begin{cases} P_i \geq 0 \\ \sum_{i=1}^{n} P_i = 1 \end{cases} \end{array}} \quad (114)$$

(Primal form of the Maximum Likelihood principle problem of modelling occurrence probabilities)

The dual form of (114) may be shown to be

$$\boxed{\underset{\{P\}}{\max}\ \underset{\{\alpha\}}{\min} \sum_{i=1}^{n} f_i \ln(P_i) - \lambda\left(\left(\sum_{i=1}^{n} P_i\right) - 1\right) + \sum_{i=1}^{n} \alpha_i P_i}, \quad (115)$$

$$=: \mathcal{L}(\{P\}, \lambda, \{\alpha\})$$

(Dual form of the Maximum Likelihood principle problem of modelling occurrence probabilities)

where λ is the Lagrange multiplier associated with the equality constraint $\sum_{i=1}^{m} P_i = 1$ and $\alpha_i \geq 0$ are the Lagrange multipliers associated with the inequality constraints $P_i \geq 0$. Problem (115) is also associated with a so-called complementary slackness condition,

$$\boxed{\alpha_i^* P_i^* = 0 \quad \forall i = 1, \dots, m} \tag{116}$$

(Complementary slackness condition)

and the constraint $\boxed{\sum_{i=1}^{m} P_i = 1}$ (117)

Note that since $\alpha_i^* \geq 0$, if $P_i^* > 0$, then (116) implies $\alpha_i^* = 0$.

Note: (Karush-Kuhn-Tucker theorem)

Theorem:

The primal form

$$\max_{\{x\}} f(\{x\}) \quad \text{subject to} \begin{cases} g_i(\{x\}) = 0 \ \forall i = 1, \dots, m, \\ h_j(\{x\}) \geq 0 \ \forall j = 1, \dots, l \end{cases} \tag{118}$$

where $\{x\} \in \mathbb{R}^n$, is equivalent to the dual form

$$\max_{\{x\}} \min_{\{\alpha\}} \mathcal{L}(\{x\}) = f(\{x\}) - \sum_{i=1}^{m} \lambda_i g_i(\{x\}) + \sum_{j=1}^{n} \alpha_j h_j(\{x\})$$
$$\text{subject to } g_i(\{x\}) = 0 \ \forall i = 1, \dots, m,$$
$$\alpha_j^* h_j(\{x\}) = 0 \ \forall j = 1, \dots, l.$$
$$\text{with } \alpha_j \geq 0. \tag{119}$$

Demonstration:

Let

$$\boxed{\mathcal{L}(\{x\}, \{\lambda\}, \{\alpha\}) = f(\{x\}) - \sum_{i=1}^{m} \lambda_i g_i(\{x\}) + \sum_{j=1}^{n} \alpha_j h_j(\{x\})} \tag{120}$$

be the Lagrangian.

Well, one may easily understand that for $h_j < 0$ (that is, in the case of the inequality constraint not being satisfied) $\alpha_j = +\infty$ would minimise $\mathcal{L}$, being this $-\infty$. If $h_j \geq 0$ (that is, in the case of the inequality being satisfied) $\alpha_j = 0$ would minimise $\mathcal{L}$, being this $f(\{x\}) - \sum_{i=1}^{m} \lambda_i g_i(\{x\})$.

Therefore,

$$\max_{\{x\}} \min_{\{\alpha\}} \mathcal{L}(\{x\}, \{\lambda\}, \{\alpha\}) = \max_{\{x\}} \begin{cases} -\infty, & \text{if } \{x\} \in \mathbb{R}^n : h_j < 0 \\ f - \sum_{i=1}^{m} \lambda_i g_i, & \text{if } \{x\} \in \mathbb{R}^n : h_j \geq 0 \end{cases}$$

$$\Leftrightarrow \boxed{\max_{\{x\}} \min_{\{\alpha\}} \mathcal{L}(\{x\}, \{\lambda\}, \{\alpha\}) = \max_{\{x\}} \left(f - \sum_{i=1}^{m} \lambda_i g_i \right), \quad \text{subject to } h_j \geq 0 \ \forall j = 1, \dots, l} \tag{121}$$

From (121), one may start to grasp the idea that (118) and (119) are equivalent.

Also from (121) one may derive the complementary slackness condition:

$$\max_{\{\alpha\}} \mathcal{L}(\{x\},\{\lambda\},\{\alpha^*\}) = \max_{\{x\}} \left(f - \sum_{i=1}^{m} \lambda_i g_i\right) \Leftarrow \text{(120)}$$

$$\Leftrightarrow f - \sum_{i=1}^{m} \lambda_i g_i + \sum_{j=1}^{r} \alpha_j^* h_j = f - \sum_{i=1}^{m} \lambda_i g_i \Leftrightarrow$$

$$\Leftrightarrow \sum_{j=1}^{r} \underbrace{\alpha_j^* h_j}_{=0 \text{ via } \alpha_j^*=0 \text{ and } h_j=0} = 0 \Leftrightarrow \boxed{\alpha_j^* h_j = 0 \;\forall j = 1,\dots,r} \quad (122)$$

(Complementary slackness condition)

To finalize the proof, one has to show that

$$\max_{\{x\}} \left(f - \sum_{i=1}^{m} \lambda_i g_i\right) = \max_{\{x\}} f \text{ subject to } g_i = 0 \;\forall i = 1,\dots,m$$

Let $\{x^*\}$ be the point pertaining to hypersurface $\mathcal{S} = \{\{x\} \in \mathbb{R}^n : g_i(\{x\}) = 0 \;\forall i = 1,\dots,m\}$, that maximizes f within it. And let $\{\varepsilon\}$ be an infinitesimal displacement of $\{x^*\}$ within $\mathcal{S}$. Then both points $\{x^*\}$ and $\{x^*\}+\{\varepsilon\}$ pertain to $\mathcal{S}$,

$$g_i(\{x^*\}) = g_i(\{x^*\}+\{\varepsilon\}) = 0 \Leftrightarrow g_i(\{x^*\}-\{\varepsilon\}) - g_i(\{x^*\}) = 0 \Leftrightarrow$$

$$\Leftrightarrow g_i(\{x^*\}) + \underbrace{\{\nabla g_i(\{x^*\})\}^T \cdot \{\varepsilon\}}_{\text{(Taylor series expansion of) first term}} + \mathcal{O}(\|\{\varepsilon\}\|^2) - g_i(\{x^*\}) = 0 \Leftrightarrow$$

$$\Leftrightarrow \boxed{\{\nabla g_i(\{x^*\})\}^T \cdot \{\varepsilon\} = 0 \;\forall i = 1,\dots,m} \quad (123)$$

$(\{\varepsilon\} \to \{0\})$

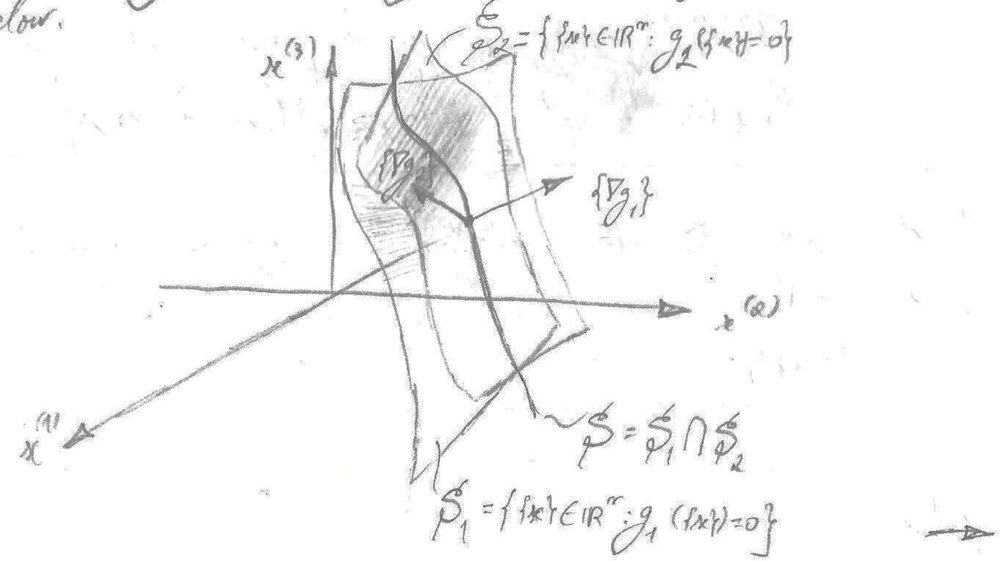
Also, since $\{x^*\}$ maximizes f within $\mathcal{S}$,

$$\lim_{\{\varepsilon\}\to\{0\}} \frac{f(\{x^*\}+\{\varepsilon\}) - f(\{x^*\})}{\|\{\varepsilon\}\|} = 0 \Leftrightarrow \text{(Taylor series expansion)}$$

$$\lim_{\{\varepsilon\}\to\{0\}} \frac{f(\{x^*\}) + \{\nabla f(\{x^*\})\}^T \cdot \{\varepsilon\} + \mathcal{O}(\|\{\varepsilon\}\|^2) - f(\{x^*\})}{\|\{\varepsilon\}\|} = 0 \Leftrightarrow$$

$$\Leftrightarrow \boxed{\lim_{\{\varepsilon\}\to\{0\}} \{\nabla f(\{x^*\})\}^T \cdot \{\varepsilon\} = 0} \quad (124)$$

From (123), one concludes that the vectors $\{\nabla g_i(\{x^*\})\}$ are the bases of the space normal to the constraint surface $\mathcal{S}$ at $\{x^*\}$, as may be intuitively understood from the picture below.



Result (124) shows the $\{\nabla f\}$ pertains to the space normal to $\mathcal{S}$ at $\{x\}=\{x^*\}$, and, therefore it is a linear combination of the base vectors $\{\nabla g_i\}$:

$$\{\nabla f(\{x^*\})\} = \sum_{i=1}^{m} \lambda_i \{\nabla g_i(\{x^*\})\} \Leftrightarrow \boxed{\{\nabla\}\left(f - \sum_{i=1}^{m} \lambda_i g_i\right)\Big|_{\{x\}=\{x^*\}} = \{0\}} \tag{125}$$

where λ_i are some scalars.

Since $\{x^*\}$ is a maximiser of f within $\mathcal{S}$, one has

$$\frac{d^2 f}{ds^2}\Big|_{\{x\}=\{x^*\}} < 0, \quad \text{where } s \text{ parametrises any curve } \Gamma \text{ within } \mathcal{S}$$

Well,

$$\frac{d^2 f}{ds^2}\Big|_{\{x\}=\{x^*\}} = \frac{d}{ds}\left(\frac{df}{ds}\right)\Big|_{\{x\}=\{x^*\}} \overset{f=f(\{x\})}{=} \frac{d}{ds}\left(\sum_{k=1}^{m} \frac{\partial f}{\partial x_k}\frac{dx_k}{ds}\right)\Big|_{\{x\}=\{x^*\}} =$$

$$= \left(\sum_{k=1}^{m}\sum_{l=1}^{m} \frac{\partial^2 f}{\partial x_k \partial x_l}\frac{dx_k}{ds}\frac{dx_l}{ds}\right)\Big|_{\{x\}=\{x^*\}} + \left(\sum_{k=1}^{m} \frac{\partial f}{\partial x_k}\frac{d^2 x_k}{ds^2}\right)\Big|_{\{x\}=\{x^*\}} \overset{(125)}{\Leftarrow}$$

$$\boxed{\frac{d^2 f}{ds^2}\Big|_{\{x\}=\{x^*\}} = \left(\sum_{k=1}^{m}\sum_{l=1}^{m} \frac{\partial^2 f}{\partial x_k \partial x_l}\frac{dx_k}{ds}\frac{dx_l}{ds}\right)\Big|_{\{x\}=\{x^*\}} + \left(\sum_{k=1}^{m}\sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_k}\frac{d^2 x_k}{ds^2}\right)\Big|_{\{x\}=\{x^*\}}} \tag{126}$$

$\frac{d^2 g_i}{ds^2} = 0$ since $g_i = 0 = \text{const}$ within Γ in $\mathcal{S}$ $\Leftrightarrow$

$$\Leftrightarrow \left(\sum_{k=1}^{m}\sum_{l=1}^{m} \frac{\partial^2 g_i}{\partial x_k \partial x_l}\frac{dx_k}{ds}\frac{dx_l}{ds}\right) + \left(\sum_{k=1}^{m} \frac{\partial g_i}{\partial x_k}\frac{d^2 x_k}{ds^2}\right) = 0 \Leftrightarrow$$

$$\Leftrightarrow \boxed{\sum_{k=1}^{m} \frac{\partial g_i}{\partial x_k}\frac{d^2 x_k}{ds^2} = -\sum_{k=1}^{m}\sum_{l=1}^{m} \frac{\partial^2 g_i}{\partial x_k x_l}\frac{dx_k}{ds}\frac{dx_l}{ds}} \tag{127}$$

From (126) and (127) one gets

$$\frac{d^2 f}{ds^2}\Big|_{\{x\}=\{x^*\}} = \left(\sum_{k=1}^{m}\sum_{l=1}^{m} \frac{dx_k}{ds}\frac{\partial^2}{\partial x_k \partial x_l}\left(f - \sum_{i=1}^{m}\lambda_i g_i\right)\frac{dx_l}{ds}\right)\Big|_{\{x\}=\{x^*\}} =$$

$$= \left(\sum_{k=1}^{m}\sum_{l=1}^{m} \frac{dx_k}{ds}\{I_k\}^T \frac{\partial^2}{\partial x_k \partial x_l}\left(f - \sum_{i=1}^{m}\lambda_i g_i\right)\{I_l\}^T \frac{dx_l}{ds}\{I_l\}\right)\Big|_{\{x\}=\{x^*\}} =$$

$$= \left(\underbrace{\left(\sum_{k=1}^{m}\frac{dx_k}{ds}\{I_k\}^T\right)}_{=\frac{d\{x\}^T}{ds}} \underbrace{\left(\sum_{k=1}^{m}\sum_{l=1}^{m}\frac{\partial^2}{\partial x_k \partial x_l}\left(f - \sum_{i=1}^{m}\lambda_i g_i\right)\{I_k\}\{I_l\}^T\right)}_{=[\mathcal{H}],\ \text{Hessian matrix of } f - \sum_{i=1}^{m}\lambda_i g_i} \underbrace{\left(\sum_{l=1}^{m}\frac{dx_l}{ds}\{I_l\}\right)}_{=\frac{d\{x\}}{ds}}\right)\Big|_{\{x\}=\{x^*\}} \Leftrightarrow$$

$$\Leftrightarrow \boxed{\frac{d^2 f}{ds^2}\Big|_{\{x\}=\{x^*\}} = \left(\frac{d\{x\}^T}{ds}[\mathcal{H}]\frac{d\{x\}}{ds}\right)\Big|_{\{x\}=\{x^*\}} = \frac{d^2}{ds^2}\left(f - \sum_{i=1}^{m}\lambda_i g_i\right)\Big|_{\{x\}=\{x^*\}}} \tag{128}$$

where $\boxed{\mathcal{H}_{kl} = \frac{\partial^2}{\partial x_k \partial x_l}\left(f - \sum_{i=1}^{m}\lambda_i g_i\right)}$ (129)

Results (125) and (128) shows that

$$\max_{\{x\}}\left(f-\sum_{i=1}^{m}\lambda_i g_i\right)=\max_{\{x\}} f \text{ subject to } g_i=0\ \forall\, i=1,\dots,m,$$

where a test on the second order derivative of $f-\sum_{i=1}^{m}\lambda_i g_i$ to access if the extremum point $\{x^*\}$ is indeed a maximum needs to be done on the parameter s that describes any curve Γ within $S=\{\{x\}\in\mathbb{R}^n : g_i(\{x\})=0\}$, that is

$$\boxed{\frac{d^2}{ds^2}\left(f-\sum_{i=1}^{m}\lambda_i g_i\right)\Big|_{\{x\}=\{x^*\}}<0} \qquad (130)$$

Note that a sufficient condition for satisfying (130) is of the Hessian matrix $[\mathcal{H}]$ given by (129) to be negative definite at $\{x\}=\{x^*\}$.

The Lagrangian in (115) is a concave function in $\{P\}$, that is

$$\frac{\partial^2\mathcal{L}}{\partial s^2}\le 0 \text{ for } s \text{ parametrising any curve in } \{P\}\in Q$$

Indeed,

$$\boxed{\frac{\partial\mathcal{L}}{\partial P_k}\underset{(115)}{=}\frac{f_k}{P_k}-\lambda+\alpha_k} \qquad (131)$$

$$\frac{\partial^2\mathcal{L}}{\partial P_k\partial P_l}\underset{(131)}{=}-\frac{f_k}{P_l^2}\delta_{kl}=\begin{cases}0 & \text{if } k\neq l\\ \underbrace{-\frac{f_k}{P_k^2}}_{\le 0} & \text{if } k=l\end{cases}\le 0$$

$$[\mathcal{H}]=\sum_{kl}\frac{\partial^2\mathcal{L}}{\partial P_k\partial P_l}\{e_k\}\{e_l\}^T=-\sum_{kl}\frac{f_k}{P_l^2}\delta_{kl}\{e_k\}\{e_l\}^T \Leftrightarrow$$

$$\Leftrightarrow \boxed{[\mathcal{H}]=-\sum_{k}\frac{f_k}{P_k^2}\{e_k\}\{e_k\}^T} \qquad (132)$$

Since $\mathcal{L}$ is a concave function in $\{P\}$, one may change the order of the maximisation-minimisation process in (115):

$$\boxed{\min_{\{\alpha\}}\max_{\{P\}}\sum_{i=1}^{n}f_i\ln(P_i)-\lambda\left(\left(\sum_{i=1}^{n}P_i\right)-1\right)+\sum_{i=1}^{n}\alpha_i P_i} \qquad (133)$$

Well,

$$\frac{\partial\mathcal{L}}{\partial P_k}\Big|_{P_k=P_k^*}=0 \overset{(131)}{\Leftrightarrow} \frac{f_k}{P_k^*}-\lambda+\alpha_k=0 \Leftrightarrow \frac{f_k-(\lambda-\alpha_k)P_k^*}{P_k^*}=0 \Leftrightarrow$$

$$\Leftrightarrow \boxed{P_k^*=\frac{f_k}{\lambda-\alpha_k} \text{ if } P_k^*\neq 0} \qquad (134)$$

Since (134) holds for $P_k^*\neq 0$, one has from the complementary slackness condition that $\alpha_k^*=0$. Therefore,

$$\boxed{P_k^*=\frac{f_k}{\lambda} \text{ if } P_k^*\neq 0} \qquad (135)$$

From (135), one concludes that

$$\lim_{f_k \to 0} P_k^* = 0 ,$$

and by continuity, one may simple consider

$$\boxed{P_k^* = \frac{f_k}{\lambda}} \quad (136)$$

By imposing the equality constraint (117), one gets

$$\sum_{i=1}^{m} P_i^* = 1 \underset{(134)}{\Leftrightarrow} \sum_{i=1}^{m} \frac{f_i}{\lambda} = 1 \Leftrightarrow \boxed{\lambda = \sum_{i=1}^{m} f_i} . \quad (137)$$

And, therefore,

$$\boxed{P_k^* = \frac{f_k}{\sum_{i=1}^{m} f_i} \overset{(108)}{=} \hat{P}(\{a_k\})} \quad (138)$$

The empirical distribution $\hat{P}(\{a\}) = f(\{a\}) / \sum_{\{\alpha\} \in \mathcal{X}}^{m} f(\{\alpha\})$ is the Maximum Likelihood estimate for the probability of occurrence.

Relative Entropy

The relative-entropy (RE) measure between two non-negative vectors $\{x\}, \{y\} \in \mathbb{R}^m$ is defined as

$$\boxed{\mathcal{D}(\{x\} \| \{y\}) = \sum_{i=1}^{m} x^{(i)} \ln\left(\frac{x^{(i)}}{y^{(i)}}\right) - \sum_{i=1}^{m} x^{(i)} + \sum_{i=1}^{m} y^{(i)}} . \quad (139)$$

(Relative-entropy measure)

If $\{x\}$ and $\{y\}$ are probability vectors, that is, pertaining to the probability simplex $\mathcal{Q}$ defined by (113), the respective RE measure is called "Kullback-Leibler divergence".

Note: (Properties of the RE measure)

The RE measure is non-symmetric, that is,

$$\boxed{\mathcal{D}(\{x\} \| \{y\}) \neq \mathcal{D}(\{y\} \| \{x\}) \text{ if } \{x\} \neq \{y\}} \quad (140)$$

(Non-symmetry of the RE measure)

as may be easily seen from the definition (139).

The RE measure is always non-negative:

$$\boxed{\mathcal{D}(\{x\} \| \{y\}) \geq 0} \quad (141)$$

(Non-negativity of the RE measure)

Indeed,

←

$$D(\{x^{(i)}\}\|\{y^{(i)}\}) \overset{(139)}{=} \underbrace{\sum_{i=1}^{m} x^{(i)} \ln\left(\frac{x^{(i)}}{y^{(i)}}\right)}_{=\left(\sum_{i=1}^{m} x^{(i)}\right) \ln\left(\frac{\sum_{i=1}^{m} x^{(i)}}{\sum_{i=1}^{m} y^{(i)}}\right),} - \sum_{i=1}^{m} x^{(i)} + \sum_{i=1}^{m} y^{(i)} \quad (\Leftarrow)$$

(Let $\bar{x} := \sum_{i=1}^{m} x^{(i)}$, $\bar{y} := \sum_{i=1}^{m} y^{(i)}$)

from the log-sum inequality

$$(\Leftarrow) \quad D(\{x^{(i)}\}\|\{y^{(i)}\}) \geq m(\bar{x} \ln\left(\frac{\bar{x}}{\bar{y}}\right) - \bar{x} + \bar{y}) \quad (\Leftarrow)$$

$\bar{x} \ln\left(\frac{\bar{x}}{\bar{y}}\right) > \bar{x} - \bar{y}$ ✓ because such $\ln(a+1) > \frac{a}{a+1}$ for $a \geq -1 \wedge a \neq 0$ ($\Rightarrow$)

$$(\Leftarrow) \ln\left(\frac{\bar{x}}{\bar{y}}\right) > \frac{\frac{\bar{x}}{\bar{y}} - 1}{\frac{\bar{x}}{\bar{y}}} \text{ for } \frac{\bar{x}}{\bar{y}} \geq 0 \wedge \frac{\bar{x}}{\bar{y}} \neq -1 \quad (\Leftarrow)$$

$\left(a+1 := \frac{\bar{x}}{\bar{y}} \ (\Rightarrow a = \frac{\bar{x}}{\bar{y}} - 1)\right)$

(There are always satisfied since $x^{(i)} \geq 0$, $y^{(i)} \geq 0$)

$$(\Leftarrow) \quad \frac{\bar{x}}{\bar{y}} \ln\left(\frac{\bar{x}}{\bar{y}}\right) > \frac{\bar{x}}{\bar{y}} - 1 \quad (\Leftarrow)$$

$$(\Leftarrow) \quad \bar{x} \ln\left(\frac{\bar{x}}{\bar{y}}\right) > \bar{x} - \bar{y}$$

One may prove $\ln(a+1) > \frac{a}{a+1}$ for $a \geq -1 \wedge a \neq 0$ by considering the following reasoning:

- Let $f(a) = \ln(a+1) - \frac{a}{a+1}$, with $a \geq -1$
- Well, $f'(a) = \frac{1}{a+1} - \frac{1}{a+1} + \frac{a}{(a+1)^2} = \frac{a}{(a+1)^2}$
- Therefore,
  $f'(a) < 0$ if $a < 0$
  $f'(a) > 0$ if $a > 0$
- $f(a)$ has thus an absolute minimum at $a = 0$.

→

- Therefore,

$$f(a) > f(0) \text{ for } a \geq -1 \wedge a \neq 0 \quad (\Leftrightarrow)$$

$$(\Leftrightarrow) \ln(a+1) - \frac{a}{a+1} > \underbrace{\ln(0+1) - \frac{0}{0+1}}_{=0} \text{ for } a \geq -1 \wedge a \neq 0 \quad (\Leftrightarrow)$$

$$(\Leftrightarrow) \ln(a+1) > \frac{a}{a+1} \text{ for } a \geq -1 \wedge a \neq 0$$

$$(\Leftrightarrow) \quad D(\{x^{(i)}\}\|\{y^{(i)}\}) \geq m(\bar{x} - \bar{y} - \bar{x} + \bar{y}) \quad (\Leftrightarrow)$$

$$D(\{x^{(i)}\}\|\{y^{(i)}\}) \geq 0. \quad \square$$

which proves (141).

The RE measure is null if and only if the respective distributions are coincident, that is,

$$\boxed{D(\{x^{(i)}\}\|\{y^{(i)}\}) = 0 \iff \{x^{(i)}\} = \{y^{(i)}\}} \tag{142}$$

One may show (142) by having the following reasoning: let $\{x^{(i)}\}$ and $\{y^{(i)}\}$ be such that $D(\{x^{(i)}\}\|\{y^{(i)}\}) = 0$. Simultaneously one does knows from the log-sum inequality that

$$0 = D(\{x^{(i)}\}\|\{y^{(i)}\}) = \sum_{i=1}^{n} x^{(i)} \ln\left(\frac{x^{(i)}}{y^{(i)}}\right) - m\bar{x} + m\bar{y} \overset{\text{(log-sum inequality)}}{\geq} m\bar{x} \ln\left(\frac{\bar{x}}{\bar{y}}\right) - m\bar{x} + m\bar{y} \geq 0$$

Due to the simultaneous equality and inequality to zero, one infers that for $\{x^{(i)}\}$ and $\{y^{(i)}\}$ that satisfy $D(\{x^{(i)}\}\|\{y^{(i)}\}) = 0$, →

$$\sum_{i=1}^{n} x^{(i)} \ln\left(\frac{x^{(i)}}{y^{(i)}}\right) - m\bar{x} + m\bar{y} = m\bar{x} \ln\left(\frac{\bar{x}}{\bar{y}}\right) - m\bar{x} + m\bar{y} \quad (\Leftrightarrow) \quad \boxed{\sum_{i=1}^{n} x^{(i)} \ln\left(\frac{x^{(i)}}{y^{(i)}}\right) = m\bar{x} \ln\left(\frac{\bar{x}}{\bar{y}}\right)} \tag{143}$$

From (143), one may further get

(143): $m\left(\sum_{i=1}^{m} y^{(i)}\right) \sum_{i=1}^{m} \frac{y^{(i)}}{\sum_{i=1}^{m} y^{(i)}} \frac{x^{(i)}}{y^{(i)}} \ln\left(\frac{x^{(i)}}{y^{(i)}}\right) = m\bar{x} \ln\left(\frac{\bar{x}}{\bar{y}}\right)$ $(\Rightarrow$

($\frac{y^{(i)}}{m\sum_{i=1}^{m} y^{(i)}} \geq 0$ and $z \ln(z)$ is a strictly convex function

($\frac{\partial^2}{\partial z^2}(z\ln(z)) = \frac{\partial}{\partial z}(\ln(z)+1) = \frac{1}{z} > 0$ for $z \geq 0$), one has

from Jensen's inequality that

$$\sum_{i=1}^{m} \frac{y^{(i)}}{m\sum_{i=1}^{m} y^{(i)}} z^{(i)} \ln(z^{(i)}) > \left(z\ln(z)\right)\Big|_{z=\sum_{i=1}^{m} \frac{y^{(i)}}{m\sum_{i=1}^{m} y^{(i)}} \frac{x^{(i)}}{y^{(i)}}}$$

- if $z^{(i)} = \frac{x^{(i)}}{y^{(i)}}$ is not uniform $(\Rightarrow$

$(\Rightarrow$ $\sum_{i=1}^{m} \frac{y^{(i)}}{m\sum_{i=1}^{m} y^{(i)}} \frac{x^{(i)}}{y^{(i)}} \ln\left(\frac{x^{(i)}}{y^{(i)}}\right) > \left(\sum_{i=1}^{m} \frac{y^{(i)}}{m\sum_{i=1}^{m} y^{(i)}} \cdot \frac{x^{(i)}}{y^{(i)}}\right) \ln\left(\sum_{i=1}^{m} \frac{y^{(i)}}{m\sum_{i=1}^{m} y^{(i)}} \cdot \frac{x^{(i)}}{y^{(i)}}\right)$

if $z^{(i)} = \frac{x^{(i)}}{y^{(i)}}$ is not uniform $(\Rightarrow$

$(\Rightarrow$ $\sum_{i=1}^{m} \frac{y^{(i)}}{m\sum_{i=1}^{m} y^{(i)}} \frac{x^{(i)}}{y^{(i)}} \ln\left(\frac{x^{(i)}}{y^{(i)}}\right) > \frac{\bar{x}}{\bar{y}} \ln\left(\frac{\bar{x}}{\bar{y}}\right)$ if $\frac{x^{(i)}}{y^{(i)}}$ is not uniform $(\Rightarrow$

$(\Rightarrow$ $m\sum_{i=1}^{m} y^{(i)} \sum_{i=1}^{m} \frac{y^{(i)}}{m\sum_{i=1}^{m} y^{(i)}} \cdot \frac{x^{(i)}}{y^{(i)}} \ln\left(\frac{x^{(i)}}{y^{(i)}}\right) > m\bar{x} \ln\left(\frac{\bar{x}}{\bar{y}}\right)$ if $\frac{x^{(i)}}{y^{(i)}}$ is not uniform $(\Rightarrow$

$(\Rightarrow$ $m\sum_{i=1}^{m} y^{(i)} \sum_{i=1}^{m} \frac{y^{(i)}}{m\sum_{i=1}^{m} y^{(i)}} \cdot \frac{x^{(i)}}{y^{(i)}} \ln\left(\frac{x^{(i)}}{y^{(i)}}\right) = m\bar{x}\ln\left(\frac{\bar{x}}{\bar{y}}\right)$ if $\frac{x^{(i)}}{y^{(i)}}$ is uniform →

$(\Rightarrow$ $\frac{x^{(i)}}{y^{(i)}}$ is uniform, that is,

$$\boxed{x^{(i)} = k y^{(i)} \quad \forall i = 1, \dots, m} \tag{144}$$

where k is some constant.

Well,

$$D(\{x\}\|\{y\}) = 0 \overset{(139)}{\Rightarrow} \sum_{i=1}^{m} x^{(i)} \ln\left(\frac{x^{(i)}}{y^{(i)}}\right) - \sum_{i=1}^{m} x^{(i)} + \sum_{i=1}^{m} y^{(i)} = 0 \Rightarrow$$

$$\overset{(144)}{\Rightarrow} k\sum_{i=1}^{m} y^{(i)} \ln\left(\frac{k y^{(i)}}{y^{(i)}}\right) - k\sum_{i=1}^{m} y^{(i)} + \sum_{i=1}^{m} y^{(i)} = 0 \Rightarrow$$

$$\Rightarrow k\ln(k)\sum_{i=1}^{m} y^{(i)} - k\sum_{i=1}^{m} y^{(i)} + \sum_{i=1}^{m} y^{(i)} = 0 \Rightarrow$$

$$\Rightarrow k(\ln(k)-1)+1 = 0 \Rightarrow \boxed{k=1} \tag{145}$$

($\frac{\partial}{\partial k}\left(k(\ln(k)-1)+1\right) = \ln(k)-1+1 = \ln(k)$

$\left.\begin{aligned} &\frac{\partial}{\partial k}\left(k(\ln(k)-1)+1\right) < 0 \text{ if } k<1 \\ &\frac{\partial}{\partial k}\left(k(\ln(k)-1)+1\right) > 0 \text{ if } k>1 \end{aligned}\right\} \Rightarrow k(\ln(k)-1)+1\Big|_{k=1} = 0$ is an absolute minimum of $k(\ln(k)-1)+1$

$k=1$ is then the only solution of $k(\ln(k)-1)+1=0$)

From (144) and (145) one concludes that $D(\{x\}\|\{y\}) = 0$ iff $\{x\} = \{y\}$.

→

←

One may also show that the RE measure is a convex function in $\{x\}$ and $\{y\}$, that is,

$$\boxed{[\mathcal{H}_{\{x\}}] \text{ where } (\mathcal{H}_{\{x\}})_{kj} = \frac{\partial^2 \mathcal{D}}{\partial x^{(k)} \partial x^{(j)}} \text{ is positive semi-definite}} \tag{146}$$

and

$$\boxed{[\mathcal{H}_{\{y\}}] \text{ where } (\mathcal{H}_{\{y\}})_{kj} = \frac{\partial^2 \mathcal{D}}{\partial y^{(k)} \partial y^{(j)}} \text{ is positive semi-definite}} \tag{147}$$

One may show (146) and (147) through the following reasoning:

$$\frac{\partial \mathcal{D}}{\partial x^{(k)}} \overset{(139)}{=} \ln\left(\frac{x^{(k)}}{y^{(k)}}\right) + 1 - 1 \Leftrightarrow \boxed{\frac{\partial \mathcal{D}}{\partial x^{(k)}} = \ln\left(\frac{x^{(k)}}{y^{(k)}}\right)} \tag{148}$$

$$\boxed{\frac{\partial^2 \mathcal{D}}{\partial x^{(k)} \partial x^{(j)}} \overset{(148)}{=} \frac{1}{x^{(k)}} \delta_{kj} \overset{x^{(k)} \geq 0}{\text{, which implies that } [\mathcal{H}_{\{x\}}] \text{ is a diagonal matrix of non-negative entries.}}} \tag{149}$$

$$\boxed{\frac{\partial \mathcal{D}}{\partial y^{(k)}} \overset{(139)}{=} -\frac{x^{(k)}}{y^{(k)}} + 1} \tag{150}$$

$$\boxed{\frac{\partial^2 \mathcal{D}}{\partial y^{(k)} \partial y^{(j)}} \overset{(150)}{=} \frac{x^{(k)}}{(y^{(k)})^2} \delta_{kj} \overset{\geq 0}{\text{, which implies the } [\mathcal{H}_{\{y\}}] \text{ is a diagonal matrix of non-negative entries.}}} \tag{151}$$

Note: (Log-sum inequality)

The log-sum inequality is such that

$$\boxed{\sum_{i=1}^{n} a_i \ln\left(\frac{a_i}{b_i}\right) \geq \left(\sum_{i=1}^{n} a_i\right) \ln\left(\frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}\right)}, \tag{152}$$

for non-negative a_i and b_i. One may show (152) through the following reasoning:

$$\sum_{i=1}^{n} a_i \ln\left(\frac{a_i}{b_i}\right) = \sum_{i=1}^{n} b_i \frac{a_i}{b_i} \ln\left(\frac{a_i}{b_i}\right) \overset{(\text{let } f(x) = x \ln(x))}{=} \sum_{i=1}^{n} b_i f\left(\frac{a_i}{b_i}\right) =$$

$$= \left(\sum_{i=1}^{n} b_i\right) \sum_{i=1}^{n} \frac{b_i}{\sum_{i=1}^{n} b_i} f\left(\frac{a_i}{b_i}\right) \Rightarrow$$

$$\left(\text{Since } \frac{b_i}{\sum_{i=1}^{n} b_i} \geq 0 \text{ and } f(x) \text{ as a convex function } \left(\frac{d^2 f}{dx} = \frac{d}{dx}(\ln(x)+1) = \frac{1}{x} \geq 0\right), \text{ Jensen's inequality holds}\right)$$

$$\Rightarrow \sum_{i=1}^{n} a_i \ln\left(\frac{a_i}{b_i}\right) \geq \left(\sum_{i=1}^{n} b_i\right) f\left(\sum_{i=1}^{n} \frac{b_i}{\sum_{i=1}^{n} b_i} \frac{a_i}{b_i}\right) \Leftrightarrow$$

$$\Leftrightarrow \sum_{i=1}^{n} a_i \ln\left(\frac{a_i}{b_i}\right) \geq \left(\sum_{i=1}^{n} b_i\right) \left(\sum_{i=1}^{n} \frac{b_i}{\sum_{i=1}^{n} b_i} \frac{a_i}{b_i}\right) \ln\left(\sum_{i=1}^{n} \frac{b_i}{\sum_{i=1}^{n} b_i} \frac{a_i}{b_i}\right) \Leftrightarrow$$

$$\Leftrightarrow \sum_{i=1}^{n} a_i \ln\left(\frac{a_i}{b_i}\right) \geq \left(\sum_{i=1}^{n} a_i\right) \ln\left(\frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}\right) \quad \square$$

Note: (Jensen's inequality)

Jensen's inequality may be stated in the following way:

Let $f(x)$ be a convex function, that is, with $\frac{d^2 f}{dx^2} \geq 0$, and $a_1, \dots, a_n \geq 0$ some non-negative coefficients. Then, for $n \geq 1$, with distinct x_i:

$$\sum_{i=1}^{n} \frac{a_i}{\sum_{i=1}^{n} a_i} f(x_i) \geq f\left(\sum_{i=1}^{n} \frac{a_i}{\sum_{i=1}^{n} a_i} x_i \right) \tag{153}$$

(Jensen's inequality for the case of f being convex)

• If $f(x)$ is strictly convex then one may change the greater or equal sign "$\geq$" in (153) to "$>$"

$$\sum_{i=1}^{n} \frac{a_i}{\sum_{i=1}^{n} a_i} f(x_i) > f\left(\sum_{i=1}^{n} \frac{a_i}{\sum_{i=1}^{n} a_i} x_i \right) \tag{154}$$

(Jensen's inequality for the case of f being strictly convex)

• If $f(x)$ is concave instead of convex, that is, with $\frac{d^2 f}{dx^2} \leq 0$, then, the greater or equal sign "$\geq$" in (153) would need to be changed to "$\leq$":

$$\sum_{i=1}^{n} \frac{a_i}{\sum_{i=1}^{n} a_i} f(x_i) \leq f\left(\sum_{i=1}^{n} \frac{a_i}{\sum_{i=1}^{n} a_i} x_i \right) \tag{155}$$

(Jensen's inequality for the case of f being concave)

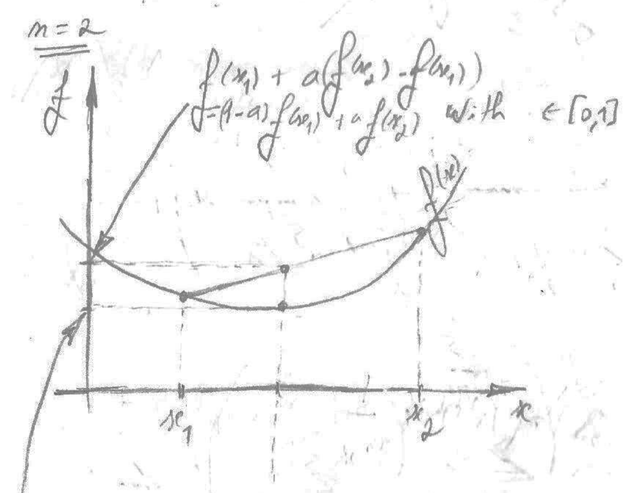
• If $f(x)$ is strictly concave, then one may change the less or equal sign "$\leq$" in (155) to "$<$":

$$\sum_{i=1}^{n} \frac{a_i}{\sum_{i=1}^{n} a_i} f(x_i) < f\left(\sum_{i=1}^{n} \frac{a_i}{\sum_{i=1}^{n} a_i} x_i \right) \tag{156}$$

(Jensen's inequality for the case of f being strictly concave)

Let us now show (153) through induction.

$n = 2$



$$x = x_1 + a(x_2 - x_1)$$
$$= (1-a)x_1 + a x_2, \text{ with } a \in [0,1]$$

$$f(x) = f((1-a)x_1 + a x_2)$$

Since f is a convex function, then,

$$(1-a) f(x_1) + a f(x_2) \geq f((1-a)x_1 + a x_2) \quad \left(\text{Let } \begin{array}{l} a_1 = 1-a \\ a_2 = a \\ \sum_{i=1}^{2} a_i = a_1 + a_2 = 1 \end{array} \right)$$

$$\Leftrightarrow \sum_{i=1}^{2} \frac{a_i}{\sum_{i=1}^{n} a_i} f(x_i) \geq f\left(\sum_{i=1}^{2} \frac{a_i}{\sum_{i=1}^{n} a_i} x_i \right)$$

The identity (152) is then proved for $n=2$.

$\underline{n}$

Let one now show that (153) holding for $n-1$ also also holds for n. Well,

$$\sum_{i=1}^{n} \frac{a_i}{\sum_{i=1}^{n} a_i} f(x_i) = \left(\sum_{i=1}^{n-1} \frac{a_i}{\sum_{i=1}^{n-1} a_i} f(x_i) \right) \cdot \frac{\sum_{i=1}^{n-1} a_i}{\sum_{i=1}^{n} a_i}$$

$$+ \frac{a_n}{\sum_{i=1}^{n} a_i} f(x_n)$$

$$\geq f\left(\sum_{i=1}^{n-1} \frac{a_i}{\sum_{i=1}^{n-1} a_i} x_i \right) \cdot \frac{\sum_{i=1}^{n-1} a_i}{\sum_{i=1}^{n} a_i} + \frac{a_n}{\sum_{i=1}^{n} a_i} f(x_n)$$

$$\geq f\left(\frac{\sum_{i=1}^{n-1} a_i}{\sum_{i=1}^{n} a_i} \left(\sum_{i=1}^{n-1} \frac{a_i}{\sum_{i=1}^{n-1} a_i} x_i \right) + \frac{a_n}{\sum_{i=1}^{n} a_i} x_n \right)$$

(f is convex)

$$\Rightarrow \sum_{i=1}^{n} \frac{a_i}{\sum_{i=1}^{n} a_i} f(x_i) \geq f\left(\left(\sum_{i=1}^{n-1} \frac{a_i}{\sum_{i=1}^{n} a_i} x_i \right) + \frac{a_n}{\sum_{i=1}^{n} a_i} x_n \right)$$

$$\Rightarrow \sum_{i=1}^{n} \frac{a_i}{\sum_{i=1}^{n} a_i} f(x_i) \geq f\left(\sum_{i=1}^{n} \frac{a_i}{\sum_{i=1}^{n} a_i} x_i \right) \quad \square$$

(Minimum RE / Maximum Likelihood Duality I)

Theorem: Let $\{f_i\}_{i=0}^{n}$ be the occurrence frequency of a training example. The empirical distribution $\{\hat{P}_i\} = \frac{\{f_i\}}{\sum_i f_i}$ is a Maximum Likelihood estimate of the occurrence probabilities iff it is the minimizer of the RE measure between $\{f_i\}$ and the probability vector $\{P_i\}$ that is,

$$\{\hat{P}_i\} = \underset{\{P_i\} \in Q}{\operatorname{argmin}} \mathcal{D}(\{f_i\} \| \{P_i\}) = \underset{\{P_i\} \in Q}{\operatorname{argmax}} \prod_{i=1}^{n} P_i^{f_i} \tag{157}$$

Proof:

$$\mathcal{D}(\{f_i\} \| \{P_i\}) \overset{(139)}{=} \sum_i f_i \ln\left(\frac{f_i}{P_i}\right) - \sum_i f_i + \underbrace{\sum_i P_i}_{=1, \text{ since } \sum_i P_i = 1}$$

$$= \sum_i f_i \ln f_i - \sum_i f_i \ln(P_i) - \sum_i f_i + 1$$

Therefore,

$$\underset{\{P_i\} \in Q}{\operatorname{argmin}} \mathcal{D}(\{f_i\} \| \{P_i\}) = \underset{\{P_i\} \in Q}{\operatorname{argmin}} \Big(-\sum_i f_i \ln(P_i) + \underbrace{\sum_i f_i \ln f_i - \sum_i f_i + 1}_{\text{does not depend on } \{P_i\}} \Big)$$

$$= \underset{\{P_i\} \in Q}{\operatorname{argmin}} \Big(-\sum_i f_i \ln(P_i) \Big) =$$

$$= \underset{\{P_i\} \in Q}{\operatorname{argmax}} \Big(\sum_i f_i \ln(P_i) \Big) = \underset{\{P_i\} \in Q}{\operatorname{argmax}} \Big(\ln\big(\prod_i P_i^{f_i} \big) \Big)$$

$$= \underset{\{P_i\} \in Q}{\operatorname{argmax}} \Big(\prod_i P_i^{f_i} \Big) \quad \square$$

(the logarithm function monotonously increases)

Note: (It is not necessary to impose constraint $\{P\} \geq 0$ when using the RE measure for obtaining the optimum vector of probabilities $\{P^*\}$)

From theorem (157), one may conclude that there is no need to impose $\{P\} \geq 0$ when finding the optimum vector of probabilities $\{P^*\}$ through minimization of the RE measure. Indeed, from (134) one had concluded that one may a priori consider the Lagrange multipliers associated with such constraint, α_k, to be null, which is equivalent to simply not taking into account the constraint. The non-consideration of the constraint in the RE measure minimization would then simplify the optimization process.

Maximum Entropy and Duality Maximum Likelihood / Maximum Entropy

Let

$$\boxed{H(\{P\}) = -\sum_{i=1}^{m} P_i \cdot \ln(P_i)}, \quad (158)$$

(Entropy function)

be the so-called entropy function.

Theorem: (Minimum RE / Maximum Entropy Duality)

$$\boxed{\underset{\{P\} \in Q}{\operatorname{argmin}} D(\{P\} \| \tfrac{1}{m}\{1\}) = \underset{\{P\} \in Q}{\operatorname{argmax}} H(\{P\})}, \quad (159)$$

where $\{1\}$ is an m-dimensional vector of ones: $\boxed{\{1\} = [1 \ \cdots \ 1]^T}$ (160)

Therefore $\frac{1}{m}\{1\}$ corresponds to an m-dimensional vector of $1/m$:

$$\boxed{\tfrac{1}{m}\{1\} = \left[\tfrac{1}{m} \ \cdots \ \tfrac{1}{m}\right]^T} \quad (161)$$

Theorem (159) states that the closest distribution of probabilities $\{P\}$ (in the sense of the first variable of RE) to the uniform one, $\frac{1}{m}\{1\}$, is the one that maximizes the entropy $H(\{P\})$.

Proof:

$$\underset{\{P\} \in Q}{\operatorname{argmin}} D(\{P\} \| \tfrac{1}{m}\{1\}) \overset{(139)}{=} \underset{\{P\} \in Q}{\operatorname{argmin}} \left(\sum_{i=1}^{m} P_i \ln\left(\frac{P_i}{\frac{1}{m}}\right) - \underbrace{\sum_{i=1}^{m} P_i}_{=1} + \underbrace{\sum_{i=1}^{m} \tfrac{1}{m}}_{=1} \right) =$$

$$= \underset{\{P\} \in Q}{\operatorname{argmin}} \Bigg(\underbrace{\sum_{i=1}^{m} P_i \ln(m)}_{=1} + \sum_{i=1}^{m} P_i \ln(P_i) \Bigg) =$$

(does not depend on $\{P\}$)

$$= \underset{\{P\} \in Q}{\operatorname{argmax}} - \sum_{i=1}^{m} P_i \ln(P_i) = \dots$$

$$\dots = \underbrace{-\sum_{i=1}^{m} P_i \ln(P_i)}_{= H(\{P\})}$$

$$= \underset{\{P\} \in Q}{\operatorname{argmax}} H(\{P\}) \quad \square$$

Note:

Note that in theorem (159), the minimisation of $\mathcal{D}(\{x\}\|\{y\})$ is with respect to the first variable $\{x\}$ while in theorem (157) the minimisation is with respect to the second variable $\{y\}$. The $\mathcal{RE}$ measure is not symmetric: even if the functions $\mathcal{D}$ in (157) and (159) were the same, their minimisation would not necessary be.

Theorem: (Minimum $\mathcal{RE}$ / Maximum Likelihood Duality II)

Let $\{P_0\} = \frac{\{f\}}{\sum_{i=1}^{n} f_i}$ be the estimate occurrence probability vector, or "prior probability". One may show that theorem (157) is equivalent to

$$\boxed{\{P^*\} = \underset{\{P\}\in Q}{\operatorname{argmin}}\ \mathcal{D}(\{P_0\}\|\{P\}) = \underset{\{P\}\in Q}{\operatorname{argmax}} \prod_{i=1}^{n} P_i^{f_i}} \tag{162}$$

Proof:

From the proof of (157) with the substitution of f_i by $\frac{f_i}{\sum_{i=1}^{n} f_i}$ one gets

$$\underset{\{P\}\in Q}{\operatorname{argmin}}\ \mathcal{D}(\{P_0\}\|\{P\}) = \underset{\{P\}\in Q}{\operatorname{argmax}} \left(\prod_{i=1}^{n} P_i^{\frac{f_i}{\sum_{i=1}^{n} f_i}}\right) =$$

$$= \underset{\{P\}\in Q}{\operatorname{argmax}} \left(\prod_{i=1}^{n} P_i^{f_i}\right)^{\frac{1}{\sum_{i=1}^{n} f_i}} \overset{f_i \geq 0}{=}$$

$$= \underset{\{P\}\in Q}{\operatorname{argmax}} \prod_{i=1}^{n} P_i^{f_i} \quad \square$$

Let one now consider the problem of finding the probability vector $\{P\}$ that is the closest (in the sense of the first variable of the $\mathcal{RE}$ measure) to a given prior $\{P_0\}$, and whose expected value of some quantity $\{a\}$ corresponds to

$$\mathbb{E}_{\{P\}}(\{a\}) = \mathbb{E}_{\{P_0\}}(\{a\}) = \sum_{i=1}^{n} \{a_i\} P_i = \sum_{i=1}^{n} \{a_i\} P_{0,i} =: \{b\} \iff$$

$$\iff \boxed{[a]\{P\} = [a]\{P_0\} =: \{b\}}, \tag{163}$$

where $a_{ji} = [a]_{ji}$ is the i-th value that the j-th entry of the vector $\{a\}$, $j = 1, \dots, k$. Also, let $\mathcal{P}$ be the set of points of the probability simplex Q (see (113)) that satisfy (163), that is

$$\boxed{\mathcal{P}(\{P_0\},[a]) = \left\{ \{P\} \in \mathbb{R}^n : \{P\} \geq 0,\ \sum_{i=1}^{n} P_i = 1,\ [a]\{P\} = [a]\{P_0\} \right\}} \tag{164}$$

(Probability simplex whose expected value of $\{a\}$ coincides with the one obtained from prior $\{P_0\}$ given the matrix of values $[a]$)

The solution of the problem is then,

$$\boxed{\{P^*\} = \underset{\{P\}\in \mathcal{P}(\{P_0\},[a])}{\operatorname{argmin}}\ \mathcal{D}(\{P\}\|\{P_0\})} \tag{165}$$

The dual form of (165) is according to the min-max counterpart of (119),

$$\{P^*\} = \underset{\{P\}}{\operatorname{argmin}}\ \underset{\{\alpha\}}{\operatorname{max}}\ \mathcal{D}(\{P\}\|\{P_0\}) - \lambda\left(\left(\sum_{i=1}^{n} P_i\right) - 1\right) - \sum_{j=1}^{k} \mu_j \sum_{i=1}^{n} a_{ji}(P_i - P_{0,i}) + \sum_{i=1}^{n} \alpha_i P_i$$

subject to
$$\begin{cases} \sum_{i=1}^{n} P_i^* = 1 \\ \sum_{i=1}^{n} [a]\{P^*\} = \sum_{i=1}^{n} [a]\{P_0\} \\ \alpha_i^* P_i^* = 0 \ \forall\, i = 1, \dots, n \end{cases}$$

with $\{\alpha\} \geq 0$.

And since the RE measure is according to (146) a convex function, one may exchange the minimize and maximize operators in (166). Let $\mathcal{L}(\{p_\ell\}, \lambda, \{\mu_j\}, \{\alpha_\ell\})$ be the respective Lagrangian. One has

$$\frac{\partial \mathcal{L}}{\partial p_\ell} = \frac{\partial}{\partial p_\ell}\left(\sum_{i=1}^{m} p_i \ln\left(\frac{p_i}{p_{0,i}}\right) - \underbrace{\sum_{i=1}^{m} p_i}_{=1} + \underbrace{\sum_{i=1}^{m} p_{0,i}}_{=1} - \lambda\left(\left(\sum_{i=1}^{m} p_i\right) - 1\right) \right.$$

$$\left. - \sum_{j=1}^{K} \mu_j \sum_{i=1}^{m} a_{ji}\,(p_i - p_{0,i}) + \sum_{i=1}^{m} \alpha_i p_i \right) \quad (\ell = 1, \dots, m)$$

$$\boxed{\frac{\partial \mathcal{L}}{\partial p_\ell} = \ln\left(\frac{p_\ell}{p_{0,\ell}}\right) + 1 - \lambda - \sum_{j=1}^{K} \mu_j a_{j\ell} + \alpha_\ell} \qquad (167)$$

$$\left.\frac{\partial \mathcal{L}}{\partial p_\ell}\right|_{p_\ell = p_\ell^*} = 0 \;\Leftrightarrow\; \ln\left(\frac{p_\ell^*}{p_{0,\ell}}\right) + 1 - \lambda - \sum_{j=1}^{K} \mu_j a_{j\ell} + \alpha_\ell = 0 \;\Leftrightarrow$$

$$\Leftrightarrow \boxed{p_\ell^* = p_{0,\ell}\, e^{-1+\lambda+\sum_{j=1}^{K} \mu_j a_{j\ell} - \alpha_\ell}} \qquad (168)$$

The complementary slackness condition $\alpha_\ell^* p_\ell^* = 0$ implies that $\alpha_\ell^* = 0$ if $p_\ell^* \neq 0$, that is,

$$\boxed{p_\ell^* = p_{0,\ell}\, e^{-1+\lambda+\sum_{j=1}^{K} \mu_j a_{j\ell}} \quad \text{if } p_\ell^* \neq 0} \qquad (169)$$

From (169), one understands that the only way of $p_\ell^* \to 0$ is by having $p_{0,\ell} \to 0$ (since having $\lambda + \sum_{j=1}^{K} \mu_j a_{j\ell} \to -\infty$ would make all p_ℓ^* to be null).

By continuity of the p_ℓ^* function, one may then write

$$\boxed{p_\ell^* = p_{0,\ell}\, e^{-1+\lambda+\sum_{j=1}^{K} \mu_j a_{j\ell}}} \qquad (170)$$

Note: (It is not necessary to impose constraint $\{p_\ell\} \geq 0$ when using the RE measure for obtaining the optimum vector of probabilities $\{p_\ell^*\}$ as the first variable of RE.

Since one had concluded that $\{\alpha_\ell\} = \{0\}$ would have been a priori considered, one may say that the constraint $\{p_\ell\} \geq 0$ would have been immediately disregarded, as it would still be satisfied through the optimization process of finding the optimum first variable of the RE measure. Such realization had already been achieved for the case of the optimum reward variable (see page 114).

By imposing the condition $\sum_{\ell=1}^{m} p_\ell^*$, one gets

$$\sum_{\ell=1}^{m} p_{0,\ell}\, e^{-1+\lambda+\sum_{j=1}^{K} \mu_j a_{j\ell}} = 1 \;\Leftrightarrow\; \left(\sum_{\ell=1}^{m} p_{0,\ell}\, e^{\sum_{j=1}^{K} \mu_j a_{j\ell}}\right) e^{-1+\lambda} = 1 \;\Leftrightarrow$$

$$\Leftrightarrow \boxed{e^{-1+\lambda} = \frac{1}{\sum_{\ell=1}^{m} p_{0,\ell}\, e^{\sum_{j=1}^{K} \mu_j a_{j\ell}}} =: \frac{1}{\tilde{Z}}}, \qquad (171)$$

where $\boxed{Z = \sum_{l=1}^{m} P_{0,l} \cdot e^{\sum_{j=1}^{k} \mu_j a_{jl}}}$ is the so-called

partition function. (172)

Equation (170) may then be re-expressed as

$$\boxed{P_l^* = \frac{P_{0,l}}{Z} e^{\sum_{j=1}^{k} \mu_j a_{jl}}, \quad l = 1, \dots, m} \quad (173)$$

(Gibbs distribution)

The coefficients μ_j with $j = 1, \dots, k$ would need to be determined through constraint (163): $[a]\{P^*\} = [a]\{P_0\}$. Such determination will not be herein explored as the solution for $\{\mu\}$ cannot be written in an explicit way.

The general expression (173) with Z given by (172) is termed Gibbs distribution. Let $\mathcal{G}$ be the set of points $\{P\}$ of the probability simplex $\mathcal{Q}$ that satisfy (173), that is, so that $\{P\}$ corresponds to a Gibbs distribution:

$$\boxed{\mathcal{G}(\{P_0\}, [a]) = \left\{ \{P\} \in \mathbb{R}^m : P_i = \frac{P_{0,i} \, e^{\sum_{j=1}^{k} \mu_j a_{ji}}}{\sum_{l=1}^{m} P_{0,l} \cdot e^{\sum_{j=1}^{k} \mu_j a_{jl}}}, \text{ with } \{\mu\} \in \mathbb{R}^k \right\}} \quad (174)$$

(Probability simplex whose points follow Gibbs distributions associated with prior $\{P_0\}$ and matrix of values $[a]$ for the quantity $f(x)$)

One may conclude that the solution of (165) corresponds to

$$\boxed{\{P^*\} = \mathcal{P}(\{P_0\}, [a]) \cap \mathcal{G}(\{P_0\}, [a])} \quad (175)$$

It is possible to show that $\{P^*\}$ also coincides with

$$\boxed{\{P^*\} = \underset{\substack{\{P\} \in \mathcal{G}(\{P_0\}, [a])}}{\operatorname{argmin}} \mathcal{D}(\{P_0\} \| \{P\})} \quad (176)$$

Indeed,

$$\mathcal{D}(\{P_0\} \| \{P\}) \overset{(139)}{=} \sum_{i=1}^{m} P_{0,i} \ln\left(\frac{P_{0,i}}{P_i}\right) - \overbrace{\sum_{i=1}^{m} P_{0,i}}^{=1} + \overbrace{\sum_{i=1}^{m} P_i}^{=1} =$$

$$= \sum_{i=1}^{m} P_{0,i} \ln\left(\frac{\sum_{l=1}^{m} P_{0,l} \cdot e^{\sum_{j=1}^{k} \mu_j a_{jl}}}{e^{\sum_{j=1}^{k} \mu_j a_{ji}}} \right) =$$

($\{P\} \in \mathcal{G}(\{P_0\}, [a])$ given by (174))

$$\mathcal{D}(\{P_0\} \| \{P\}) = \underbrace{\left(\sum_{i=1}^{m} P_{0,i}\right)}_{=1} \ln\left(\sum_{l=1}^{m} P_{0,l} \cdot e^{\sum_{j=1}^{k} \mu_j a_{jl}}\right) - \sum_{i=1}^{m} \sum_{j=1}^{k} \mu_j a_{ji} P_{0,i} \Rightarrow$$

$$\Rightarrow \boxed{\mathcal{D}(\{P_0\} \| \{P\}) = \ln\left(\sum_{l=1}^{m} P_{0,l} \cdot e^{\sum_{j=1}^{k} \mu_j a_{jl}}\right) - \sum_{i=1}^{m} \sum_{j=1}^{k} \mu_j a_{ji} P_{0,i}} \quad (177)$$

$$\frac{\partial \mathcal{D}}{\partial \mu_m} = \frac{\sum_{l=1}^{m} P_{0,l}\, a_{ml} \cdot e^{\sum_{j=1}^{k} \mu_j a_{jl}}}{\sum_{l=1}^{m} P_{0,l} \cdot e^{\sum_{j=1}^{k} \mu_j a_{jl}}} - \sum_{i=1}^{m} a_{mi} P_{0,i} =$$

$$= \sum_{l=1}^{m} a_{ml} \underbrace{\frac{P_{0,l} \cdot e^{\sum_{j=1}^{k} \mu_j a_{jl}}}{\sum_{l=1}^{m} P_{0,l} \cdot e^{\sum_{j=1}^{k} \mu_j a_{jl}}}}_{= P_l \text{ from (174)}} - \sum_{i=1}^{m} a_{mi} P_{0,i} (=)$$

$$(=) \boxed{\frac{\partial \mathcal{D}}{\partial \mu_m} = \sum_{i=1}^{m} a_{mi} P_i - \sum_{i=1}^{m} a_{mi} P_{0,i}} \quad (178)$$

$$\frac{\partial \mathcal{D}}{\partial \mu_m}\Big|_{\mu_m = \mu_m^*} = 0 \;(\Leftrightarrow \sum_{i=1}^{m} a_{mi} P_i^* = \sum_{i=1}^{m} a_{mi} P_{0,i} \;(\Leftrightarrow$$

$$(\Leftrightarrow \boxed{[a]\{P^*\} = [a]\{P_0\}}, \quad (179)$$

which shows (176).

One may thus enunciate the following theorem:

Theorem:

The following are equivalent:

- (165): $\boxed{\{P^*\} = \underset{\{P\} \in \mathcal{P}(\{P_0\}, [a])}{\operatorname{argmin}} \mathcal{D}(\{P\} \| \{P_0\})}$ (180)
- (176): $\boxed{\{P^*\} = \underset{\{P\} \in \mathcal{Y}(\{P_0\}, [a])}{\operatorname{argmin}} \mathcal{D}(\{P_0\} \| \{P\})}$ (181)
- (175): $\boxed{\{P^*\} = \mathcal{P}(\{P_0\}, [a]) \cap \mathcal{Y}(\{P_0\}, [a])}$ (182)

Note that (181) is according to (157) a Minimum Relative Entropy / Maximum Likelihood duality. And if $\{P_0\} = \frac{1}{n}\{1\}$, (180) is according to (159) a Minimum Relative Entropy / Maximum Entropy duality. In such conditions, the equivalence of (180) and (181) may then be termed a Maximum Likelihood / Maximum Entropy duality.

IV- Expectation Maximisation Algorithm: Maximum Likelihood over Mixture of Distributions

The Expectation Maximisation (EM) Algorithm: General

Let $\boxed{\mathcal{D} = \{x_1, x_2, \dots, x_m\}}$ be a set of training data.

Also let y be some "hidden" (or "latent") random variable. (183)

that in some (a priori unknown way influences the training set.

Let $\{y\} = [y_1, \dots y_m]^T$ be the set of y values associated with $\mathcal{D}$. Briefly,

Let $\{\theta\}$ be the set of parameters of the model that may

describe the training set and the hidden variable.

The Maximum Likelihood problem, (61), would be to find the set of parameters $\{\theta\}$ that maximizes the conditional probability $P(\mathcal{D}|\{\theta\})$, that is,

$$\{\theta^*\} = \underset{\{\theta\}}{\operatorname{argmax}}\ P(\mathcal{D}|\{\theta\}) = \underset{\{\theta\}}{\operatorname{argmax}}\ \log\left(P(\mathcal{D}|\{\theta\})\right) \quad (*) \leftarrow (49)$$

(logarithm monotonously increase)

$$\boxed{\{\theta^*\} = \underset{\{\theta\}}{\operatorname{argmax}}\ \underbrace{\log\left(\sum_{\{y\}} P(\mathcal{D}, \{y\}|\{\theta\})\right)}_{=: L(\{\theta\})}} \quad (184)$$

Function $L(\{\theta\})$ that needs to be maximized is quite complex: it corresponds to the logarithm of the sum of other functions. It would then be more convenient to transform it into a sum of logarithms, hence the very existence of the EM algorithm

Well,

$$L(\{\theta\}) = \log\left(\sum_{\{y\}} P(\mathcal{D}, \{y\}|\{\theta\})\right) = \tag{185}$$

$$= \boxed{L(\{\theta\}) = \log\left(\sum_{\{y\}} q(\{y\}|\mathcal{D},\{\theta\}) \cdot \frac{P(\mathcal{D},\{y\}|\{\theta\})}{q(\{y\}|\mathcal{D},\{\theta\})}\right)}$$

← Let $q(\{y\}|\mathcal{D},\{\theta\})$ be some distribution of the set $\{y\}$ conditioned to the training set $\mathcal{D}$ and the set of parameters $\{\theta\}$, not necessarily $P(\{y\}|\mathcal{D},\{\theta\})$

Since $q(\{y\}|\mathcal{D},\{\theta\}) \Rightarrow \sum_{\{y\}} q(\{y\}|\mathcal{D},\{\theta\}) = 1$ and the logarithm is a concave function, one may use Jensen's inequality (155) and write

$$\boxed{L(\{\theta\}) \geq \underbrace{\sum_{\{y\}} q(\{y\}|\mathcal{D},\{\theta\}) \log\left(\frac{P(\mathcal{D},\{y\}|\{\theta\})}{q(\{y\}|\mathcal{D},\{\theta\})}\right)}_{=: Q(q,\{\theta\})}} \tag{186}$$

In (186), one defines a weighted sum of logarithms, $Q(q,\{\theta\})$, as a lower bound for a logarithm of a weighted sum, $L(\{\theta\})$. The EM algorithm is such that function Q is maximized step-by-step which according to (186) would become a value that is less or equal than the actual maximum of $L(\{\theta\})$. The algorithm consists of

- defining initial estimates for $\{\theta\}$ as $\{\theta^{(0)}\}$
- iterating the following two-steps until convergence of $\{\theta\}$:
  - Maximization of Q with respect to q: $\boxed{q^{(t+1)} = \underset{q}{\operatorname{argmax}}\, Q(q, \{\theta^{(t)}\})}$ (187)
  - Maximization of Q with respect to $\{\theta\}$: $\boxed{\{\theta^{(t+1)}\} = \underset{\{\theta\}}{\operatorname{argmax}}\, Q(q^{(t+1)}, \{\theta\})}$ (188)

The EM algorithm expressed by (187) and (188) may be shown to result in a maximum of Q that actually coincides with the maximum of L. Such proof would require the usage of the so-called Jordan-Bishop's theorem:

Theorem: (Jordan-Bishop's theorem)

The optimum q at the $t+1$-th step is $P(\{y\}|\mathcal{D},\{\theta^{(t)}\})$ that is,

$$\boxed{q^{(t+1)} = \underset{q}{\operatorname{argmax}}\, Q(q,\{\theta^{(t)}\}) = P(\{y\}|\mathcal{D},\{\theta^{(t)}\})} \quad \text{("Expectation step")} \tag{189}$$

Proof:

Since according to (186), $L(\{\theta^{(t)}\}) = Q(q, \{\theta^{(t)}\})$, if one shows that $L(\{\theta^{(t)}\}) = Q(q^{(t+1)}, \{\theta^{(t)}\}) = Q(P(\{y\}|\mathcal{D},\{\theta^{(t)}\}), \{\theta^{(t)}\})$ then one would also show that (189) would maximise $Q(q,\{\theta^{(t)}\})$.

Well,

$$Q(q,\{\theta^{(t)}\})\Big|_{q=P(\{y\}|\mathcal{D},\{\theta^{(t)}\})} = \sum_{\{y\}} P(\{y\}|\mathcal{D},\{\theta^{(t)}\}) \log\left(\frac{P(\mathcal{D},\{y\}|\{\theta^{(t)}\})}{P(\{y\}|\mathcal{D},\{\theta^{(t)}\})}\right) =$$

$$\overset{(52)}{=} \sum_{\{y\}} P(\{y\}|\mathcal{D},\{\theta^{(t)}\}) \log\left(\frac{P(\{y\}|\mathcal{D},\{\theta^{(t)}\}) \cdot P(\mathcal{D}|\{\theta^{(t)}\})}{P(\{y\}|\mathcal{D},\{\theta^{(t)}\})}\right) =$$

$$= \left(\underbrace{\sum_{\{y\}} P(\{y\}|\mathcal{D},\{\theta^{(t)}\})}_{=1 \text{ from (55)}}\right) \log\left(P(\mathcal{D}|\{\theta^{(t)}\})\right) =$$

$$= \log\left(P(\mathcal{D}|\{\theta^{(t)}\})\right) = \log\left(\sum_{\{y\}} P(\mathcal{D},\{y\}|\{\theta^{(t)}\})\right) = \rightarrow$$

$$\Leftarrow \quad \boxed{Q\left(\underbrace{P(\{y\}|\mathcal{D},\{\theta^{(t)}\})}_{=q^{(t+1)}},\{\theta^{(t)}\}\right)=\log\left(\sum_{\{y\}}P(\mathcal{D},\{y\}|\{\theta^{(t)}\})\right)=L(\{\theta^{(t)}\})} \tag{190}$$

□

Now if one substitutes (189) in to the rhs and of (188), one gets

$$Q(q^{(t+1)},\{\theta\}) = Q(P(\{y\}|\mathcal{D},\{\theta^{(t)}\}),\{\theta\}) =$$

$$\overset{(186)}{=}\sum_{\{y\}}P(\{y\}|\mathcal{D},\{\theta^{(t)}\})\log\left(\frac{P(\mathcal{D},\{y\}|\{\theta\})}{P(\{y\}|\mathcal{D},\{\theta^{(t)}\})}\right) \Leftarrow$$

$$\Leftarrow \quad \boxed{\begin{aligned}Q(q^{(t+1)},\{\theta\}) = &\sum_{\{y\}}P(\{y\}|\mathcal{D},\{\theta^{(t)}\})\log\left(P(\mathcal{D},\{y\}|\{\theta\})\right)\\ &-\sum_{\{y\}}P(\{y\}|\mathcal{D},\{\theta^{(t)}\})\log\left(P(\{y\}|\mathcal{D},\{\theta^{(t)}\})\right)\end{aligned}} \tag{191}$$

Since the second term in (191) does not depend on $\{\theta\}$, one may disregard it in the maximisation of $Q(q^{(t+1)},\{\theta\})$ at step (188), and write

$$\boxed{\{\theta^{(t+1)}\} = \underset{\{\theta\}}{\operatorname{argmax}}\sum_{\{y\}}P(\{y\}|\mathcal{D},\{\theta^{(t)}\})\log\left(P(\mathcal{D},\{y\}|\{\theta\})\right)} \tag{192}$$

("Maximisation step")

The (189) is termed "expectation step", since it allows one to construct the operand of the maximisation operator of (192) and this operand is the expectation value of $\log(P(\mathcal{D},\{y\}|\{\theta\}))$ with respect to a distribution of $\{y\}$ whose probability corresponds to $P(\{y\}|\mathcal{D},\{\theta^{(t)}\})$, that is,

$$\boxed{\sum_{\{y\}}P(\{y\}|\mathcal{D},\{\theta^{(t)}\})\log\left(P(\mathcal{D},\{y\}|\{\theta\})\right) = \mathop{\mathbb{E}}_{\{y\}\sim P(\{y\}|\mathcal{D},\{\theta^{(t)}\})}\log\left(P(\mathcal{D},\{y\}|\{\theta\})\right)} \tag{193}$$

The (192) is termed "maximise for step", since, trivially, it maximises the expectation with respect to $\{\theta\}$.

The maximisation step allows one to get $\{\theta\}=\{\theta^{(t+1)}\}$ that maximises the expectation. The expectation would then be greater or equal to $L(\{\theta^{(t)}\})$ but less or equal than $L(\{\theta^{(t+1)}\})$. The subsequent expectation step would then retrieve the value of $L(\{\theta^{(t+1)}\})$. By iterating on it, convergence at a maximum of L would be attained.

The EM algorithm for the case of i.i.d. data

Let one now consider that the data pairs $\{x_1\}, \dots, \{x_m\}$ are i.i.d between themselves, as well as $y_1 \dots y_m$ and $(\{x_1\}, y_1), \dots, (\{x_m\}, y_m)$. This means that

$$P(\mathcal{D}|\{\theta\}) = P(\{x_1\}, \dots, \{x_m\}|\{\theta\}) \overset{(65)}{=} \boxed{P(\mathcal{D}|\{\theta\}) = \prod_{i=1}^{m} P(\{x_i\}|\{\theta\})} \quad (194)$$

$$P(\mathcal{D}, \{y\}|\{\theta\}) = P(\{x_1\}, \dots, \{x_m\}, y_1, \dots, y_m|\{\theta\}) \overset{(194)}{=}$$

$$= \boxed{P(\mathcal{D}, \{y\}|\{\theta\}) = \prod_{i=1}^{m} P(\{x_i\}, y_i|\{\theta\})} \quad (195)$$

$$\cdot\ P(\{y\}|\mathcal{D}, \{\theta\}) \overset{(52)}{=} \frac{P(\mathcal{D}, \{y\}|\{\theta\})}{P(\mathcal{D}|\{\theta\})} \overset{(194)\ (195)}{=} \frac{\prod_{i=1}^{m} P(\{x_i\}, y_i|\{\theta\})}{\prod_{i=1}^{m} P(\{x_i\}|\{\theta\})} \overset{(52)}{=}$$

$$= \frac{\prod_{i=1}^{m} P(y_i|\{x_i\}, \{\theta\})\, P(\{x_i\}|\{\theta\})}{\prod_{i=1}^{m} P(\{x_i\}|\{\theta\})} \overset{(52)}{=}$$

$$\overset{(52)}{=} \boxed{P(\{y\}|\mathcal{D}, \{\theta\}) = \prod_{i=1}^{m} P(y_i|\{x_i\}, \{\theta\})} \quad (196)$$

The expectation step (189) now becomes (196)

$$\boxed{q^{(t+1)} = \arg\max_{q} Q(q, \{\theta^{(t)}\}) = \prod_{i=1}^{m} P(y_i|\{x_i\}, \{\theta^{(t)}\})} \quad (197)$$

("Expectation step", in the case of $\{x_1\}, \dots, \{x_m\}$, as well as $y_1, \dots, y_m$ and $(\{x_1\}, y_1), \dots, (\{x_m\}, y_m)$ being i.i.d.)

The maximization step (192) becomes

$$\{\theta^{(t+1)}\} = \arg\max_{\{\theta\}} \sum_{\{y\}} P(\{y\}|\mathcal{D}, \{\theta^{(t)}\}) \log\left(P(\mathcal{D}, \{y\}|\{\theta\})\right) \overset{(196)}{=}$$

$$= \arg\max_{\{\theta\}} \sum_{\{y\}} \prod_{i=1}^{m} P(y_i|\{x_i\}, \{\theta^{(t)}\}) \log\left(\prod_{j=1}^{m} P(\{x_j\}, y_j|\{\theta\})\right) =$$

(Let $\boxed{y_i \in \{\alpha_1, \dots, \alpha_k\}}$ (198))

$$= \arg\max_{\{\theta\}} \sum_{y_1 \in \{\alpha_1, \dots, \alpha_k\}} \dots \sum_{y_m \in \{\alpha_1, \dots, \alpha_k\}} \prod_{i=1}^{m} P(y_i|\{x_i\}, \{\theta^{(t)}\}) \times$$

$$\times \sum_{j=1}^{m} \log\left(P(\{x_j\}, y_j|\{\theta\})\right) \overset{(52)}{=}$$

$$\Leftrightarrow \{\theta^{(t+1)}\} = \underset{\{\theta\}}{\arg\max} \left(\sum_{y_1 \in \{\alpha_1,\dots,\alpha_k\}} P(y_1 | \{x_1\}, \{\theta^{(t)}\}) \times \log\left(P(\{x_1\}, y_1 | \{\theta\})\right)\right) \times$$

$$\times \left(\sum_{y_2 \in \{\alpha_1,\dots,\alpha_k\}} P(y_2 | \{x_2\}, \{\theta^{(t)}\})\right) \times \dots \times \left(\sum_{y_m \in \{\alpha_1,\dots,\alpha_k\}} P(y_m | \{x_m\}, \{\theta^{(t)}\})\right)$$

$$+ \left(\sum_{y_1 \in \{\alpha_1,\dots,\alpha_k\}} P(y_1 | \{x_1\}, \{\theta^{(t)}\})\right) \times \left(\sum_{y_2 \in \{\alpha_1,\dots,\alpha_k\}} P(y_2 | \{x_2\}, \{\theta^{(t)}\}) \times \log\left(P(\{x_2\}, y_2 | \{\theta\})\right)\right) \times$$

$$\times \left(\sum_{y_3 \in \{\alpha_1,\dots,\alpha_k\}} P(y_3 | \{x_3\}, \{\theta^{(t)}\})\right) \times \dots \times \left(\sum_{y_m \in \{\alpha_1,\dots,\alpha_k\}} P(y_m | \{x_m\}, \{\theta^{(t)}\})\right)$$

$$+ \dots +$$

$$+ \left(\sum_{y_1 \in \{\alpha_1,\dots,\alpha_k\}} P(y_1 | \{x_1\}, \{\theta^{(t)}\})\right) \times \dots \times \left(\sum_{y_{m-1} \in \{\alpha_1,\dots,\alpha_k\}} P(y_{m-1} | \{x_{m-1}\}, \{\theta^{(t)}\})\right)$$

$$\times \left(\sum_{y_m \in \{\alpha_1,\dots,\alpha_k\}} P(y_m | \{x_m\}, \{\theta^{(t)}\}) \log\left(P(\{x_m\}, y_m | \{\theta\})\right)\right) \Leftrightarrow$$

$$\Leftrightarrow \{\theta^{(t+1)}\} = \underset{\{\theta\}}{\arg\max} \sum_{y_1 \in \{\alpha_1,\dots,\alpha_k\}} P(y_1 | \{x_1\}, \{\theta^{(t)}\}) \log\left(P(\{x_1\}, y_1 | \{\theta\})\right) +$$

$$+ \dots +$$

$$+ \sum_{y_m \in \{\alpha_1,\dots,\alpha_k\}} P(y_m | \{x_m\}, \{\theta^{(t)}\}) \log\left(P(\{x_m\}, y_m | \{\theta\})\right) \Leftrightarrow$$

$$\Leftrightarrow \boxed{\{\theta^{(t+1)}\} = \underset{\{\theta\}}{\arg\max} \sum_{j=1}^{K} \sum_{i=1}^{m} P(y_i = \alpha_j | \{x_i\}, \{\theta^{(t)}\}) \log\left(P(\{x_i\}, y_i = \alpha_j | \{\theta\})\right)} \quad (199)$$

("Maximization step", in the case of $\{x_1\}, \dots, \{x_m\}$, as well as $y_1, \dots, y_m$ and $(\{x_1\}, y_1), \dots, (\{x_m\}, y_m)$ being i.i.d. and $y_i \in \{\alpha_1, \dots, \alpha_k\}$)

Example: random tossing of two coins

Let one consider the exercise of randomly picking one of two coins, tossing it three times, and repeating the picking and tossing processes $m-1$ times. Let y_i be the hidden variable associated with the i-th picking of a coin: $y_i = 1$ if coin "1" is picked and $y_i = 2$ if coin "2" is picked. The respective probabilities correspond to

$$\boxed{P(y_i = 1) = \lambda}, \quad (201) \qquad \boxed{P(y_i = 2) = 1 - \lambda} \quad (202)$$

Therefore, $\boxed{\{\alpha\} = [1 \ 2]^T}$ (203). One may write $\boxed{P(y_i = \alpha_j) = \lambda \delta_{j1} + (1-\lambda)\delta_{j2}}$ (204)

Let $\{x_i\} = [x_i^{(1)} x_i^{(2)} x_i^{(3)}]^T$ be the variable associated with the 3 tossings in the i-th turn: $x_i^{(l)} = 0$ if "heads" occurs at the l-th tossing and $x_i^{(l)} = 1$ if "tails" occurs instead. Therefore, $\{x_i\} \in \{0,1\}^3$. The probability of getting "heads" in the case of coin "1" being picked is λ. And in the case of coin "2" being picked, the probability is π. Therefore,

$$P(x_i^{(l)} = 0 \mid y_i = 1) = \lambda \tag{205}$$

$$P(x_i^{(l)} = 0 \mid y_i = 2) = \pi \tag{206}$$

One has $x_i^{(l)} \in \{\beta_1, \beta_2\}$ (207) with $\beta_1 = 0$, $\beta_2 = 1$ (208). Therefore,

$$P(x_i^{(l)} = \beta_s \mid y_i = \alpha_j) = (\lambda \delta_{s0} + (1-\lambda)\delta_{s1})\delta_{j1} + (\pi\delta_{s0} + (1-\pi)\delta_{s1})\delta_{j2} \tag{209}$$

$$P(\{x_i\} = [\beta_{s^{(1)}}, \beta_{s^{(2)}}, \beta_{s^{(3)}}]^T \mid y_i = \alpha_j) = \prod_{l=1}^{3} P(x_i^{(l)} = \beta_{s^{(l)}} \mid y_i = \alpha_j) \overset{(209)}{=}$$

($x_i^{(1)}, x_i^{(2)}, x_i^{(3)}$ are i.i.d.)

$$\overset{(=)}{=} P(\{x_i\} = [\beta_{s^{(1)}}, \beta_{s^{(2)}}, \beta_{s^{(3)}}]^T \mid y_i = \alpha_j) = \left(\prod_{l=1}^{3} (\lambda \delta_{s^{(l)},0} + (1-\lambda)\delta_{s^{(l)},1})\right)\delta_{j1} + \left(\prod_{l=1}^{3} (\pi \delta_{s^{(l)},0} + (1-\pi)\delta_{s^{(l)},1})\right)\delta_{j2} \tag{210}$$

Let $m_i = |\{l : x_i^{(l)} = 0, l = 1,2,3\}|$ (211) be the number of "heads" occurring in the i-th turn. From (210), one may understand that

$$P(\{x_i\} = [\beta_{s^{(1)}}, \beta_{s^{(2)}}, \beta_{s^{(3)}}]^T \mid y_i = \alpha_j) \overset{\text{(Abuse of notation)}}{=} P(\{x_i\} \mid y_i = \alpha_j) =$$

$$P(\{x_i\} \mid y_i = \alpha_j) = \lambda^{m_i}(1-\lambda)^{3-m_i}\delta_{j1} + \pi^{m_i}(1-\pi)^{3-m_i}\delta_{j2} \tag{212}$$

Let us now focus on the operand of the summation operator in (199). One understands that the parameters of the problem are λ, λ and π. Therefore,

$$\{\theta\} = [\lambda \;\; \lambda \;\; \pi]^T \tag{213}$$

Well,

$$P(y_i = \alpha_j \mid \{x_i\}, \{\theta\}) = \frac{P(\{x_i\} \mid y_i = \alpha_j, \{\theta\})\, P(y_i = \alpha_j, \{\theta\})}{P(\{x_i\} \mid \{\theta\})} =$$

(Bayes' theorem, (56))

$$= \frac{\lambda^{m_i}(1-\lambda)^{3-m_i}\delta_{j1} + \pi^{m_i}(1-\pi)^{3-m_i}\delta_{j2}}{\sum_{j=1}^{2} P(\{x_i\}, y_i = \alpha_j \mid \{\theta\})} \times$$

(212), (204), (49)

$$\times (\lambda\delta_{j1} + (1-\lambda)\delta_{j2}) \overset{(52)}{=}$$

$$= \frac{\lambda\lambda^{m_i}(1-\lambda)^{3-m_i}\delta_{j1} + (1-\lambda)\pi^{m_i}(1-\pi)^{3-m_i}\delta_{j2}}{\sum_{j=1}^{2} P(\{x_i\} \mid y_i = \alpha_j, \{\theta\})\, P(y_i = \alpha_j \mid \{\theta\})} \overset{(=)}{=}$$

$$\Leftrightarrow P(y_i=\alpha_j \mid \{x_i\},\theta) = \frac{\lambda \chi^{m_i}\cdot(1-\chi)^{3-m_i}\delta_{j1} + (1-\lambda)\pi^{m_i}\cdot(1-\pi)^{3-m_i}\delta_{j2}}{\sum_{j=1}^{2}\lambda \chi^{m_i}\cdot(1-\chi)^{3-m_i}\delta_{j1} + (1-\lambda)\pi^{m_i}\cdot(1-\pi)^{3-m_i}\delta_{j2}}$$

(212), (204)

$$\Leftrightarrow \boxed{P(y_i=\alpha_j \mid \{x_i\},\theta) = \frac{\lambda \chi^{m_i}\cdot(1-\chi)^{3-m_i}\delta_{j1} + (1-\lambda)\pi^{m_i}\cdot(1-\pi)^{3-m_i}\delta_{j2}}{\lambda \chi^{m_i}\cdot(1-\chi)^{3-m_i} + (1-\lambda)\pi^{m_i}(1-\pi)^{3-m_i}}} \quad (214)$$

Let one now define

$$\boxed{\mu_{i,1}^{(t)} := P(y_i=1 \mid \{x_i\},\theta^{(t)}) = \frac{\lambda^{(t)}(\chi^{(t)})^{m_i}\cdot(1-\chi^{(t)})^{3-m_i}}{\lambda^{(t)}(\chi^{(t)})^{m_i}\cdot(1-\chi^{(t)})^{3-m_i} + (1-\lambda^{(t)})(\pi^{(t)})^{m_i}\cdot(1-\pi^{(t)})^{3-m_i}}}$$

(21

Therefore,

$$\boxed{P(y_i=\alpha_j \mid \{x_i\},\theta^{(t)}) = \mu_{i,1}^{(t)}\delta_{j1} + (1-\mu_{i,1}^{(t)})\delta_{j2}} \quad (216)$$

Now,

$$P(\{x_i\}, y_i=\alpha_j \mid \{\theta\}) \overset{(5.2)}{=} P(\{x_i\} \mid y_i=\alpha_j, \{\theta\})\, P(y_i=\alpha_j \mid \{\theta\}) \overset{(212),(204)}{=}$$

$$\Leftrightarrow \boxed{P(\{x_i\}, y_i=\alpha_j \mid \{\theta\}) = \lambda \chi^{m_i}\cdot(1-\chi)^{3-m_i}\delta_{j1} + (1-\lambda)\pi^{m_i}\cdot(1-\pi)^{3-m_i}\delta_{j2}} \quad (217)$$

Let one now find $\{\theta\}$ that maximizes the record of (199) at step $t+1$. Well,

$$\boxed{\sum_{j=1}^{K}\sum_{i=1}^{m} P(y_i=\alpha_j \mid \{x_i\},\theta^{(t)}) \log\left(P(\{x_i\}, y_i=\alpha_j \mid \{\theta\})\right) \overset{(216),(217)}{=}}$$

$$\boxed{\sum_{i=1}^{m}\mu_{i,1}^{(t)}\log\left(\lambda\chi^{m_i}\cdot(1-\chi)^{3-m_i}\right) + (1-\mu_{i,1}^{(t)})\log\left((1-\lambda)\pi^{m_i}\cdot(1-\pi)^{3-m_i}\right) := \mathcal{L}(\{\theta\},\{\theta^{(t)}\})} \quad (218)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \frac{\sum_{i=1}^{m}\mu_{i,1}^{(t)}}{\lambda} - \frac{\sum_{i=1}^{m}(1-\mu_{i,1}^{(t)})}{1-\lambda} = \frac{(1-\lambda)\left(\sum_{i=1}^{m}\mu_{i,1}^{(t)}\right) - \lambda m + \lambda\left(\sum_{i=1}^{m}\mu_{i,1}^{(t)}\right)}{\lambda(1-\lambda)} \Leftrightarrow$$

$$\boxed{\frac{\partial \mathcal{L}}{\partial \lambda} = \frac{\left(\sum_{i=1}^{m}\mu_{i,1}^{(t)}\right) - \lambda m}{\lambda(1-\lambda)}} \quad (219)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda}\Big|_{\lambda=\lambda^{(t+1)}} = 0 \Leftrightarrow \sum_{i=1}^{m}\mu_{i,1}^{(t)} - \lambda^{(t+1)}\cdot m = 0 \;\wedge\; \lambda^{(t+1)} \notin \{0,1\} \Leftrightarrow$$

$$\Leftrightarrow \boxed{\lambda^{(t+1)} = \frac{1}{m}\sum_{i=1}^{m}\mu_{i,1}^{(t)}} \quad (220)$$

$$\frac{\partial \mathcal{L}}{\partial \chi} = \sum_{i=1}^{m}\mu_{i,1}^{(t)}\frac{\lambda\left(m_i\chi^{m_i-1}\cdot(1-\chi)^{3-m_i} - \chi^{m_i}(3-m_i)(1-\chi)^{3-m_i-1}\right)}{\lambda\chi^{m_i}\cdot(1-\chi)^{3-m_i}} =$$

$$= \sum_{i=1}^{m}\mu_{i,1}^{(t)}\left(\frac{m_i}{\chi} - \frac{3-m_i}{1-\chi}\right) = \sum_{i=1}^{m}\mu_{i,1}^{(t)}\frac{m_i(1-\chi)-(3-m_i)\chi}{\chi(1-\chi)} \Leftrightarrow$$

$$\Leftrightarrow \frac{\partial f}{\partial \lambda} = \sum_{i=1}^{m} \mu_{i,1}^{(t)} \frac{m_i - 3\lambda}{\lambda(1-\lambda)} \Leftrightarrow \boxed{\frac{\partial f}{\partial \lambda} = \frac{\left(\sum_{i=1}^{m} m_i \mu_{i,1}^{(t)}\right) - 3\lambda\left(\sum_{i=1}^{m} \mu_{i,1}^{(t)}\right)}{\lambda(1-\lambda)}} \tag{221}$$

$$\left.\frac{\partial f}{\partial \lambda}\right|_{\lambda=\lambda^{(t+1)}} = 0 \Leftrightarrow \left(\sum_{i=1}^{m} m_i \mu_{i,1}^{(t)}\right) - 3\lambda^{(t+1)}\left(\sum_{i=1}^{m} \mu_{i,1}^{(t)}\right) = 0 \wedge \lambda^{(t+1)} \notin \{0,1\} \Leftrightarrow$$

$$\Leftrightarrow \boxed{\lambda^{(t+1)} = \frac{\sum_{i=1}^{m} \frac{m_i}{3} \mu_{i,1}^{(t)}}{\sum_{i=1}^{m} \mu_{i,1}^{(t)}}} \tag{222}$$

$$\frac{\partial f}{\partial \pi} = \sum_{i=1}^{m} (1-\mu_{i,1}^{(t)}) \frac{(1-\lambda)\left(m_i \pi^{m_i-1}(1-\pi)^{3-m_i} - (3-m_i)\pi^{m_i}(1-\pi)^{3-m_i-1}\right)}{(1-\lambda)\pi^{m_i}\cdot(1-\pi)^{3-m_i}} =$$

$$= \sum_{i=1}^{m} (1-\mu_{i,1}^{(t)}) \left(\frac{m_i}{\pi} - \frac{3-m_i}{1-\pi}\right) = \sum_{i=1}^{m} (1-\mu_{i,1}^{(t)}) \frac{m_i(1-\pi)-(3-m_i)\pi}{\pi(1-\pi)}$$

$$= \sum_{i=1}^{m} (1-\mu_{i,1}^{(t)}) \frac{m_i - 3\pi}{\pi(1-\pi)} \Leftrightarrow$$

$$\Leftrightarrow \boxed{\frac{\partial f}{\partial \pi} = \frac{\left(\sum_{i=1}^{m} m_i\left(1-\mu_{i,1}^{(t)}\right)\right) - 3\pi\left(\sum_{i=1}^{m}\left(1+\mu_{i,1}^{(t)}\right)\right)}{\pi(1-\pi)}} \tag{223}$$

$$\left.\frac{\partial f}{\partial \pi}\right|_{\pi=\pi^{(t+1)}} = 0 \Leftrightarrow \left(\sum_{i=1}^{m} \frac{m_i}{3}\left(1-\mu_{i,1}^{(t)}\right)\right) - 3\pi^{(t+1)}\left(\sum_{i=1}^{m} \mu_{i,1}^{(t)}\right) = 0 \wedge \pi^{(t+1)} \notin \{0,1\} \Leftrightarrow$$

$$\Leftrightarrow \boxed{\pi^{(t+1)} = \frac{\sum_{i=1}^{m} \frac{m_i}{3}\left(1-\mu_{i,1}^{(t)}\right)}{\sum_{i=1}^{m}\left(1-\mu_{i,1}^{(t)}\right)}} \tag{224}$$

Let one show that $\lambda^{(t+1)}$, $\lambda^{(t+1)}$ and $\pi^{(t+1)}$ indeed maximize $f(\theta^{(t)}, \theta^{(t+1)})$. (219) Well,

$$\frac{\partial^2 f}{\partial \lambda^2} = \frac{-m\lambda(1-\lambda) - \left(\left(\sum_{i=1}^{m} \mu_{i,1}^{(t)}\right) - \lambda m\right)((1-\lambda)-\lambda)}{\lambda^2(1-\lambda)^2} =$$

$$= \frac{-m\lambda(1-\lambda) - (1-2\lambda)\left(\sum_{i=1}^{m} \mu_{i,1}^{(t)}\right) + m\lambda(1-\lambda) - m\lambda^2}{\lambda^2(1-\lambda)^2} \Leftrightarrow$$

$$\Leftrightarrow \boxed{\frac{\partial^2 f}{\partial \lambda^2} = -\frac{(1-2\lambda)\left(\sum_{i=1}^{m} \mu_{i,1}^{(t)}\right) + m\lambda^2}{\lambda^2(1-\lambda)^2}} \tag{225}$$

$$\left.\frac{\partial^2 f}{\partial \lambda^2}\right|_{\lambda=\lambda^{(t+1)}} \overset{(220)}{=} -\frac{\left(1-2\lambda\frac{1}{m}\left(\sum_{i=1}^{m} \mu_{i,1}^{(t)}\right)\right)\left(\sum_{i=1}^{m} \mu_{i,1}^{(t)}\right) + \frac{1}{m}\left(\sum_{i=1}^{m} \mu_{i,1}^{(t)}\right)^2}{\left(\frac{1}{m}\sum_{i=1}^{m} \mu_{i,1}^{(t)}\right)^2\left(\frac{1}{m}\sum_{i=1}^{m}\left(1-\mu_{i,1}^{(t)}\right)\right)^2} =$$

$$= -\frac{\left(\sum_{i=1}^{m} \mu_{i,1}^{(t)}\right) - \frac{1}{m}\left(\sum_{i=1}^{m} \mu_{i,1}^{(t)}\right)^2}{\left(\frac{1}{m}\sum_{i=1}^{m} \mu_{i,1}^{(t)}\right)^2\left(\frac{1}{m}\sum_{i=1}^{m}\left(1-\mu_{i,1}^{(t)}\right)\right)^2} \Leftrightarrow$$

$$\left.\frac{\partial^2 f}{\partial \lambda^2}\right|_{\lambda=\lambda^{(t+1)}} = -\frac{m\left(\frac{1}{m}\sum_{i=1}^{m}\mu_{i,1}^{(t)}\right)\left(\frac{1}{m}\sum_{i=1}^{m}(1-\mu_{i,1}^{(t)})\right)}{\left(\frac{1}{m}\sum_{i=1}^{m}\mu_{i,1}^{(t)}\right)^2\left(\frac{1}{m}\sum_{i=1}^{m}(1-\mu_{i,1}^{(t)})\right)^2} \overset{(=)}{}$$

$$\boxed{\left.\frac{\partial^2 f}{\partial \lambda^2}\right|_{\lambda=\lambda^{(t+1)}} = -\frac{m}{\left(\frac{1}{m}\sum_{i=1}^{m}\mu_{i,1}^{(t)}\right)\left(\frac{1}{m}\sum_{i=1}^{m}(1-\mu_{i,1}^{(t)})\right)} < 0} \quad (226)$$

$$\frac{\partial^2 f}{\partial \lambda \partial \pi} = \frac{\partial^2 f}{\partial \lambda \partial \pi} = 0$$

$$\frac{\partial^2 f}{\partial f^2} \overset{(221)}{=} \frac{-3\left(\sum_{i=1}^{m}\mu_{i,1}^{(t)}\right)f(1-f) - \left(\left(\sum_{i=1}^{m} m_i \mu_{i,1}^{(t)}\right) - 3f\left(\sum_{i=1}^{m}\mu_{i,1}^{(t)}\right)\right)(1-f) - f}{f^2(1-f)^2}$$

$$\boxed{\frac{\partial^2 f}{\partial f^2} = -\frac{(1-2f)\left(\sum_{i=1}^{m} m_i \mu_{i,1}^{(t)}\right) + 3f^2\left(\sum_{i=1}^{m}\mu_{i,1}^{(t)}\right)}{f^2(1-f)^2}} \quad (227)$$

$$\left.\frac{\partial^2 f}{\partial \pi^2}\right|_{f=f^{(t+1)}} \overset{(222)}{=} -\frac{\left(1-2\frac{\sum_{i=1}^{m}\frac{m_i}{3}\mu_{i,1}^{(t)}}{\sum_{i=1}^{m}\mu_{i,1}^{(t)}}\right)\left(\sum_{i=1}^{m} m_i \mu_{i,1}^{(t)}\right) + 3\left(\frac{\sum_{i=1}^{m}\frac{m_i}{3}\mu_{i,1}^{(t)}}{\sum_{i=1}^{m}\mu_{i,1}^{(t)}}\right)^2\left(\sum_{i=1}^{m}\mu_{i,1}^{(t)}\right)}{\left(\frac{\sum_{i=1}^{m}\frac{m_i}{3}\mu_{i,1}^{(t)}}{\sum_{i=1}^{m}\mu_{i,1}^{(t)}}\right)^2\left(1-\frac{\sum_{i=1}^{m}\frac{m_i}{3}\mu_{i,1}^{(t)}}{\sum_{i=1}^{m}\mu_{i,1}^{(t)}}\right)^2}$$

$$\left.\frac{\partial^2 f}{\partial f^2}\right|_{f=f^{(t+1)}} = -\frac{3\left(\sum_{i=1}^{m}\mu_{i,1}^{(t)}\right)}{f^{(t+1)}\left(\frac{\sum_{i=1}^{m}\frac{m_i}{3}\mu_{i,1}^{(t)}}{\sum_{i=1}^{m}\mu_{i,1}^{(t)}}\right)\cdot\left(\frac{\sum_{i=1}^{m}\left(1-\frac{m_i}{3}\right)\mu_{i,1}^{(t)}}{\sum_{i=1}^{m}\mu_{i,1}^{(t)}}\right)} < 0 \quad (m_i \le 3)$$

(228)

$$\frac{\partial^2 f}{\partial f \partial \pi} = 0$$

$$\frac{\partial^2 f}{\partial \pi^2} \overset{(223)}{=} \frac{-3\left(\sum_{i=1}^{m}\left(1-\mu_{i,1}^{(t)}\right)\right)\pi(1-\pi) - \left(\left(\sum_{i=1}^{m} m_i\left(1-\mu_{i,1}^{(t)}\right)\right) - 3\pi\left(\sum_{i=1}^{m}\left(1-\mu_{i,1}^{(t)}\right)\right)\right)((1-\pi)-\pi)}{\pi^2(1-\pi)^2}$$

$$\boxed{\frac{\partial^2 f}{\partial \pi^2} = -\frac{(1-2\pi)\left(\sum_{i=1}^{m} m_i\left(1-\mu_{i,1}^{(t)}\right)\right) + 3\pi^2\left(\sum_{i=1}^{m}\left(1-\mu_{i,1}^{(t)}\right)\right)}{\pi^2(1-\pi)^2}} \quad (229)$$

$$\left.\frac{\partial^2 \mathcal{L}}{\partial \pi^2}\right|_{\pi=\pi^{(t+1)}} \overset{(224)}{=} -\frac{3\left(\sum_{i=1}^{m}(1-\mu_{i,1}^{(t)})\right)}{\left(\frac{\sum_{i=1}^{m}\frac{n_i}{3}(1-\mu_{i,1}^{(t)})}{\sum_{i=1}^{m}(1-\mu_{i,1}^{(t)})}\right)\cdot\left(\frac{\sum_{i=1}^{m}(1-\frac{n_i}{3})(1-\mu_{i,1}^{(t)})}{\sum_{i=1}^{m}(1-\mu_{i,1}^{(t)})}\right)} < 0 \quad \leftarrow n_i \leq 3 \tag{230}$$

One concludes that at $\{\theta\} = \{\theta^{(t+1)}\}$ the Hessian terms $\frac{\partial^2 \mathcal{L}}{\partial \lambda^2}$, $\frac{\partial^2 \mathcal{L}}{\partial \mu^2}$ and $\frac{\partial^2 \mathcal{L}}{\partial \pi^2}$ are all negative and the non-diagonal ones $\frac{\partial^2 \mathcal{L}}{\partial \lambda \partial \mu}$, $\frac{\partial^2 \mathcal{L}}{\partial \lambda \partial \pi}$ and $\frac{\partial^2 \mathcal{L}}{\partial \mu \partial \pi}$ are null. Hessian matrix is then negative definite at that point, and, therefore, that point is a maximizant of the function.

Gaussian Mixture Models (GMMs)

Consider the case of variable $\{x\}$ being continuous, and that there are k possible Gaussian distributions to which it may pertain, that is

$$p(\{x\} \mid y=j) = \frac{1}{\sqrt{(2\pi)^m |[\Sigma_j]|}} e^{-\frac{1}{2}(\{x\}-\{\mu_j\})^T [\Sigma_j]^{-1} (\{x\}-\{\mu_j\})} \tag{231}$$

(j-th Gaussian probability density function)

Note that the j-th Gaussian PDF described by (231) corresponds to (79) with mean vector $\{\mu_j\}$ and covariance $[\Sigma_j]$.

Let $\mathcal{D} = \{\{x_1\}, \dots, \{x_m\}\}$ (232) be the set of training data, and $\{y\} = [y_1 \dots y_m]$ (233) the vector of hidden variables. Also, let

$$\lambda_j := P(y=j), \tag{234}$$

be the probability of variable y corresponding to j. Let $\{\theta\} = [\{\theta_j\}, j=1,\dots,k]^T$ (235) with $\{\theta_j\} = [\lambda_j, \{\mu_j\}, [\Sigma_j]]^T$ (236) be the vector of parameters. One may show that in similarity to the discrete case, the Expectation Maximization Algorithm for the case of $\{x\}$ being continuous is associated with the steps

$$q^{(t+1)} = \underset{q}{\operatorname{argmax}}\, Q(q \mid \{\theta^{(t)}\}) = \prod_{i=1}^{m} P(y_i \mid \{x_i\}, \{\theta^{(t)}\}) \tag{237}$$

("Expectation step" for the case of $\{x\}$ being continuous)

$$\boxed{\{\theta^{(t+1)}\} = \underset{\{\theta\}}{\operatorname{argmax}} \sum_{j=1}^{k} \sum_{i=1}^{m} P(y_i = j \mid \{x_i\}, \{\theta^{(t)}\}) \log\left(L(\{x_i\}, y_i = j \mid \{\theta\})\right)} \tag{238}$$

("Maximization step" for the case of $\{x\}$ being continuous)

Note that in (237) and (238) it was assumed the $\{x_1\}, \ldots, \{x_m\}$ are fixed, as well as $y_1, \ldots, y_m$ and $(\{x_1\}, y_1), \ldots, (\{x_m\}, y_m)$.

Well, let

$$\boxed{w_{ij}^{(t)} := P(y_i = j \mid \{x_i\}, \{\theta\})} \tag{239}$$

(Probability of the i-th data point being associated with the j-th Gaussian distribution, given the values of the data point $\{x_i\}$ and the vector of parameters $\{\theta\}$)

One has

$$w_{ij} = P(y_i = j \mid \{x_i\}, \{\theta\}) \overset{\text{(Bayes' theorem, (56))}}{=} \frac{L(\{x_i\} \mid y_i = j \mid \{\theta\}) \cdot \overbrace{P(y_i = j \mid \{\theta\})}^{= \lambda_j, \text{ from (234)}}}{\underbrace{L(\{x_i\} \mid \{\theta\})}_{= \sum_{j=1}^{k} L(\{x_i\}, y_i = j \mid \{\theta\}), \text{ from (49)}}} = \overset{\text{(231)}}{\leftarrow}$$

$$= \frac{\frac{\lambda_j}{\sqrt{(2\pi)^m \lvert[\Sigma_j]\rvert}} e^{-\frac{1}{2}(\{x_i\} - \{\mu_j\})^T [\Sigma_j]^{-1} (\{x_i\} - \{\mu_j\})}}{\sum_{j=1}^{k} L(\{x_i\}, y_i = j \mid \{\theta\})} \overset{\text{(52)}}{\leftarrow}$$

$$\boxed{w_{ij} = \frac{\frac{\lambda_j}{\sqrt{(2\pi)^m \lvert[\Sigma_j]\rvert}} e^{-\frac{1}{2}(\{x_i\} - \{\mu_j\})^T [\Sigma_j]^{-1} (\{x_i\} - \{\mu_j\})}}{\sum_{j=1}^{k} \frac{\lambda_j}{\sqrt{(2\pi)^m \lvert[\Sigma_j]\rvert}} e^{-\frac{1}{2}(\{x_i\} - \{\mu_j\})^T [\Sigma_j]^{-1} (\{x_i\} - \{\mu_j\})}}} \tag{240}$$

Also,

$$L(\{x_i\}, y_i = j \mid \{\theta\}) \overset{(52)}{=} L(\{x_i\} \mid y_i = j, \{\theta\}) \overbrace{P(y_i = j \mid \{\theta\})}^{= \lambda_j \text{ from (234)}} \overset{\text{(231)}}{\leftarrow}$$

$$\boxed{L(\{x_i\}, y_i = j \mid \{\theta\}) = \frac{\lambda_j}{\sqrt{(2\pi)^m \lvert[\Sigma_j]\rvert}} e^{-\frac{1}{2}(\{x_i\} - \{\mu_j\})^T [\Sigma_j]^{-1} (\{x_i\} - \{\mu_j\})}} \tag{241}$$

By taking into account that equation (238) would need to be solved considering the constraint $\boxed{\sum_{j=1}^{k} \lambda_j = 1}$, (242) it would be transformed into the following dual form:

$$\boxed{\{\theta^{(t+1)}\} = \underset{\{\theta\}}{\operatorname{argmax}} \left(\sum_{j=1}^{k} \sum_{i=1}^{m} w_{ij}^{(t)} \left(\log\left(\frac{\lambda_j}{\sqrt{(2\pi)^m \lvert[\Sigma_j]\rvert}} \right) - \frac{1}{2} (\{x_i\} - \{\mu_j\})^T [\Sigma_j]^{-1} (\{x_i\} - \{\mu_j\}) \right) - b\left(\left(\sum_{j=1}^{k} \lambda_j \right) - 1 \right) \right)}$$

subject to

$$\sum_{j=1}^{k} \lambda_j = 1$$

(242)

where b is a Lagrange multiplier.

Equation (242) may also be expressed as

$$\{\theta^{(t+1)}\} = \underset{\{\theta\}}{\operatorname{argmax}}\ \mathcal{L}(\{\theta\}, \{\theta^{(t)}\}) \quad \text{subject to} \quad \sum_{j=1}^{K} \lambda_j = 1 \tag{243}$$

where,

$$\mathcal{L}(\{\theta\}, \{\theta^{(t)}\}) = \sum_{j=1}^{K}\sum_{i=1}^{nn} w_{ij}^{(t)} \left(\log\left(\frac{\lambda_j}{\sqrt{(2\pi)^m |[\Sigma_j]|}} \right) - \frac{1}{2} (\{x_i\} - \{\mu_j\})^T [\Sigma_j]^{-1} (\{x_i\} - \{\mu_j\}) \right) - \eta \left(\left(\sum_{j=1}^{K} \lambda_j \right) - 1 \right) \tag{244}$$

is the Lagrangian.

$$\frac{\partial \mathcal{L}}{\partial \lambda_l} = \left(\sum_{i=1}^{nn} \frac{w_{il}^{(t)}}{\lambda_l} \right) - \eta \quad (\Leftrightarrow) \quad \boxed{\frac{\partial \mathcal{L}}{\partial \lambda_l} = \frac{\left(\sum_{i=1}^{nn} w_{il}^{(t)} \right) - \eta \lambda_l}{\lambda_l}} \tag{245}$$

$$\left. \frac{\partial \mathcal{L}}{\partial \lambda_l} \right|_{\lambda_l = \lambda_l^{(t+1)}} = 0 \ (\Leftrightarrow) \left(\sum_{i=1}^{nn} w_{il}^{(t)} \right) - \eta \lambda_l^{(t+1)} = 0 \ \wedge \lambda_l^{(t+1)} \neq 0 \ (\Leftrightarrow) \boxed{\lambda_l^{(t+1)} = \frac{1}{\eta} \sum_{i=1}^{nn} w_{il}^{(t)}} \tag{246}$$

Let us impose condition (242) on (246):

$$\sum_{j=1}^{K} \lambda_j^{(t+1)} = 1 \ (\Leftrightarrow) \ \frac{1}{\eta} \sum_{i=1}^{nn} \underbrace{\sum_{j=1}^{K} w_{ij}^{(t)}}_{=1} = 1 \ (\Leftrightarrow) \boxed{\eta = nn} \tag{247}$$

(246)

Therefore, one may write (246) as

$$\boxed{\lambda_j^{(t+1)} = \frac{1}{nn} \sum_{i=1}^{nn} w_{ij}^{(t)}} \tag{248}$$

$$\frac{\partial \mathcal{L}}{\partial \mu_l^{(n)}} = -\frac{1}{2} \sum_{j=1}^{K} \sum_{i=1}^{nn} w_{ij}^{(t)} \frac{\partial}{\partial \mu_l^{(n)}} \left((\{x_i\} - \{\mu_j\})^T \sum_{m=1}^{m} \sum_{n=1}^{n} [\Sigma_j]^{-1}_{mn} \{e_m\} \{e_n\}^T (\{x_i\} - \{\mu_j\}) \right) =$$

$$= -\frac{1}{2} \sum_{j=1}^{K} \sum_{i=1}^{nn} w_{ij}^{(t)} \frac{\partial}{\partial \mu_l^{(n)}} \left(\sum_{m=1}^{m} \sum_{n=1}^{n} [\Sigma_j]^{-1}_{mn} (x_i^{(m)} - \mu_j^{(m)}) (x_i^{(n)} - \mu_j^{(n)}) \right) =$$

$$= -\frac{1}{2} \sum_{j=1}^{K} \sum_{i=1}^{nn} w_{ij}^{(t)} \left(\sum_{m=1}^{m} \sum_{n=1}^{n} [\Sigma_j]^{-1}_{mn} \left(-(x_i^{(m)} - \mu_j^{(m)}) \delta_{jl} \delta_{nn} - (x_i^{(n)} - \mu_j^{(n)}) \delta_{jl} \delta_{nn} \right) \right) =$$

$$= -\frac{1}{2} \sum_{i=1}^{nn} w_{il}^{(t)} \left(\left(\sum_{n=1}^{n} [\Sigma_l]^{-1}_{nn} \left(-(x_i^{(n)} - \mu_l^{(n)}) \right) \right) + \left(\sum_{m=1}^{m} [\Sigma_l]^{-1}_{mn} \left(-(x_i^{(m)} - \mu_l^{(m)}) \right) \right) \right) \ (\Leftarrow \text{ } [\Sigma_l]^{-1} \text{ is symmetric})$$

$$(\Leftrightarrow) \boxed{\frac{\partial \mathcal{L}}{\partial \mu_l^{(n)}} = \{1_n\}^T [\Sigma_l]^{-1} \left(\sum_{i=1}^{nn} w_{il}^{(t)} (\{x_i\} - \{\mu_l\}) \right)} \tag{249}$$

$$\left. \frac{\partial \mathcal{L}}{\partial \mu_l^{(n)}} \right|_{\mu_l^{(n)} = \mu_l^{(n)(t+1)}} = 0 \ (\Leftrightarrow) \quad \{1_n\}^T [\Sigma_l]^{-1} \left(\sum_{i=1}^{nn} w_{il}^{(t)} (\{x_i\} - \{\mu_l^{(t+1)}\}) \right) = 0 \ \forall n = 1, \dots, m \ (\Leftrightarrow)$$

$$(\Leftrightarrow) [\Sigma_l]^{-1} \left(\sum_{i=1}^{nn} w_{il}^{(t)} (\{x_i\} - \{\mu_l^{(t+1)}\}) \right) = \{0\} \ (\Leftarrow \text{ } \left(\begin{array}{c} \text{The null space of} \\ [\Sigma_l]^{-1} \text{ is } \{0\} \text{ since it} \\ \text{is invertible} \end{array} \right)$$

$$(\Leftrightarrow) \boxed{\{\mu_l^{(t+1)}\} = \frac{1}{\sum_{i=1}^{nn} w_{il}^{(t)}} \left(\sum_{i=1}^{nn} w_{il}^{(t)} \{x_i\} \right)} \tag{250}$$

$$\frac{\partial \mathcal{L}}{\partial \Sigma_j^{(l,n)}} = \sum_{j=1}^{K}\sum_{i=1}^{m} w_{ij}^{(t)} \left(\frac{-\frac{1}{2}\left|[\Sigma_j]\right|^{-\frac{3}{2}} \frac{\partial \left|[\Sigma_j]\right|}{\partial \Sigma_j^{(l,n)}}}{\left|[\Sigma_j]\right|^{-\frac{1}{2}}} - \frac{1}{2}\left(\{x_i\}-\{\mu_j\}\right)^T \frac{\partial [\Sigma_j]^{-1}}{\partial \Sigma_j^{(l,n)}} \left(\{x_i\}-\{\mu_j\}\right) \right)$$

$$= \sum_{i=1}^{m} w_{ij}^{(t)} \left(-\frac{1}{2}\frac{\left|[\Sigma_j]\right|}{\left|[\Sigma_j]\right|} [\Sigma_j]^{-1T}_{l,n} \right.$$

(90) (92)

$$\left. + \frac{1}{2}\left(\{x_i\}-\{\mu_j\}\right)^T \left(\sum_{m=1}^{m} [\Sigma_j]^{-1}_{ml} \{e_m\} \right) \cdot \left(\sum_{N=1}^{m} \{e_N\}^T [\Sigma_j]^{-1}_{Nn} \right) \times \right.$$

$$\left. \times \left(\{x_i\}-\{\mu_j\}\right) \right) = \quad \leftarrow \left([\Sigma_j]^{-1} \text{ is symmetric}\right)$$

$$= \sum_{i=1}^{m} w_{ij}^{(t)} \left(-\frac{1}{2}[\Sigma_j]^{-1}_{ln} + \frac{1}{2}\sum_{m=1}^{m}\sum_{N=1}^{m} (x_i^{(m)} - \mu_j^{(m)}) [\Sigma_j]^{-1}_{ml} [\Sigma_j]^{-1}_{Nn} (x_i^{(N)} - \mu_j^{(N)}) \right)$$

$$= \frac{1}{2}\sum_{i=1}^{m} w_{ij}^{(t)} \left(\overbrace{\left(\{x_i\}-\{\mu_j\}\right)^T [\Sigma_j]^{-1}}^{\text{scalars}} \{e_l\} \overbrace{\{e_n\}^T [\Sigma_j]^{-1} \left(\{x_i\}-\{\mu_j\}\right)}^{\text{scalars}} \right.$$

$$\left. - \{e_l\}^T [\Sigma_j]^{-1} \{e_n\} \right) =$$

$$= \frac{1}{2}\cdot\sum_{i=1}^{m} w_{ij}^{(t)} \left(\{e_l\}^T [\Sigma_j]^{-1} \left(\{x_i\}-\{\mu_j\}\right)\left(\{x_i\}-\{\mu_j\}\right)^T [\Sigma_j]^{-1} \{e_n\} \right.$$

$$\left. - \{e_l\}^T [\Sigma_j]^{-1} \{e_n\} \right) \Leftarrow$$

$\left([\Sigma_j]^{-1} \text{ is symmetric}\right)$

$$\Leftarrow \boxed{\frac{\partial \mathcal{L}}{\partial \Sigma_j^{(l,n)}} = \{e_l\}^T [\Sigma_j]^{-1} \left(\sum_{i=1}^{m} \frac{1}{2} w_{ij}^{(t)} \left(\left(\{x_i\}-\{\mu_j\}\right)\left(\{x_i\}-\{\mu_j\}\right)^T [\Sigma_j]^{-1} - [I] \right) \right) \{e_n\}} \quad (251)$$

$$\left.\frac{\partial \mathcal{L}}{\partial \Sigma_j^{(l,n)}}\right|_{\Sigma_j^{(l,n)} = \Sigma_j^{(l,n)(t+1)}} = 0 \quad \forall\, l,n = 1,\dots,n \quad \Leftrightarrow$$

$$\Leftrightarrow \left[\Sigma_j^{(t+1)}\right]^{-1} \left(\sum_{i=1}^{m} w_{ij}^{(t)} \left(\left(\{x_i\}-\{\mu_j\}\right)\left(\{x_i\}-\{\mu_j\}\right)^T \left[\Sigma_j^{(t+1)}\right]^{-1} - [I] \right) \right) = [0] \Leftrightarrow$$

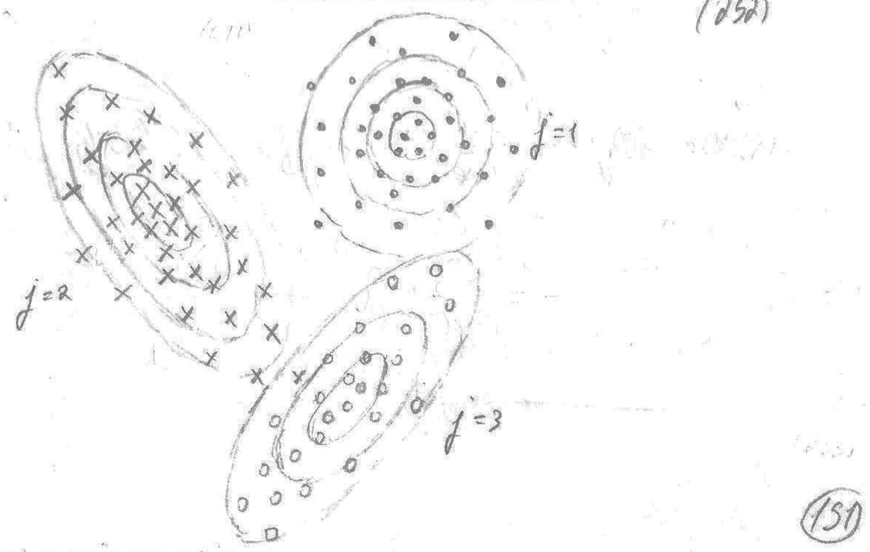
$$\Leftrightarrow \left(\sum_{i=1}^{m} w_{ij}^{(t)} \left(\{x_i\}-\{\mu_j\}\right)\left(\{x_i\}-\{\mu_j\}\right)^T \left[\Sigma_j^{(t+1)}\right]^{-1} \right) - \left(\sum_{i=1}^{m} w_{ij}^{(t)} \right)[I] = [0] \Leftrightarrow$$

(The null space of $\left[\Sigma_j^{(t+1)}\right]^{-1}$ is $\{0\}$, since $\left[\Sigma_j^{(t+1)}\right]^{-1}$ is invertible)

$$\Leftrightarrow \left(\sum_{i=1}^{m} w_{ij}^{(t)} \right)\left[\Sigma_j^{(t+1)}\right] = \sum_{i=1}^{m} w_{ij}^{(t)} \left(\{x_i\}-\{\mu_j\}\right)\left(\{x_i\}-\{\mu_j\}\right)^T \Leftrightarrow$$

(Multiplying by $\left[\Sigma_j^{(t+1)}\right]$)

$$\Leftrightarrow \boxed{\left[\Sigma_j^{(t+1)}\right] = \frac{\sum_{i=1}^{m} w_{ij}^{(t)} \left(\{x_i\}-\{\mu_j\}\right)\left(\{x_i\}-\{\mu_j\}\right)^T}{\sum_{i=1}^{m} w_{ij}^{(t)}}} \quad (252)$$



Multinomial Mixture and "bag of words" Application

Let one consider the case of m documents associated with variables $d_1, \dots, d_m$. Also let one consider n words, associated with variables $w_1, \dots, w_n$. Each document may be regarded as a "bag of words": G_{ab} would be the number of occurrences of the a-th word in the b-th document, regardless of the structure of the text in it: hence the structure (grammar, order of words, etc.) is not accounted and the document is simply taken as a set of words of some multiplicity.

A problem would be to associate a "topic" – the hidden variable $y \in \{1, \dots, K\}$ – to documents and words.

Let $\boxed{\{x^{(i)}\} = (w(i), d(i))}$ (253) be the i-th example: a couple in which the first entry $w(i) \in \{1, \dots, n\}$ is the index of the word picked in the example, and the second entry $d(i) \in \{1, \dots, m\}$ is the index of the document. One may write

$$\boxed{w(i) = a : w_a = w_{w(i)} \text{ for } a = 1, \dots n} \quad (253)$$

$$\boxed{b(i) = b : d_b = d_{d(i)} \text{ for } b = 1, \dots, m} \quad (254)$$

The examples are $\{x^{(1)}\}, \dots \{x^{(\ell)}\}$. An example is obtained by arbitrarily picking a document and a word from it.

Let $\boxed{\lambda_j = P(y=j)}$ (254) be the probability of variable y being j. Since it is a probability, it would need to satisfy

$$\boxed{\sum_{j=1}^{K} \lambda_j = 1} \quad (255)$$

Non-negativity does not need to be imposed, as it is obtained "by force" through the EM algorithm.

Let $\boxed{\mu_{ja} = P(w_a \mid y=j)}$ (256) be the probability of the a-th word occurring given the j-th topic. And let $\boxed{\nu_{jb} = P(d_b \mid y=j)}$ (257) be the probability of the b-th document occurring given the j-th topic. Since these are conditional probabilities they need to satisfy (55), that is,

$$\boxed{\sum_{a=1}^{n} \mu_{ja} = 1 \ \forall j = 1, \dots, K} \quad (258)$$

$$\boxed{\sum_{b=1}^{m} \nu_{jb} = 1 \ \forall j = 1, \dots, K} \quad (259)$$

Assuming that words and documents are conditionally independent given the topic, that is, $w \perp d \mid y$, one may show that parameters λ_j, $\{\mu_{ja}\}$ and $\{\nu_{jb}\}$ with $j = 1, \dots, K$ fully describe the problem, since these also fully describe the joint probabilities $P(w, d, y)$ and from the joint probabilities all the other may be obtained.

$$P(w=w_a, d=d_b, y=j) \overset{(52)}{=} P(w=w_a, d=d_b \mid y=j) \cdot \underbrace{P(y=j)}_{=\lambda_j} =$$

$$\overset{(w \perp d \mid y)}{\underset{(67)}{=}} \underbrace{P(w=w_a \mid y=j)}_{=\mu_{ja}} \cdot \underbrace{P(d=d_b \mid y=j)}_{=\nu_{jb}} \cdot \lambda_j =$$

$$\boxed{P(w=w_a, d=d_b, y=j) = \mu_{ja} \cdot \nu_{jb} \cdot \lambda_j} \quad (260)$$

The vector of parameters of the EM algorithm is thus

$$\boxed{\{\theta\} = [\lambda_1, \dots, \lambda_K, \mu_{11}, \dots, \mu_{Kn}, \nu_{11}, \dots \nu_{Km}]^T} \quad (261)$$

Let

$$\boxed{w_{ij} := P(y_i = j \mid \{x_i\}, \{\theta\})} \tag{262}$$

(Probability of the word and document of the i-th example being associated with the j-th topic)

One may also infer that

$$\boxed{\sum_{w(i)=n} w_{ij} = \sum_{i:\, w(i)=n} w_{ij} = \sum_{i=1}^{l} w_{ij}\, \delta_{w(i),n}} \tag{263}$$

(Probability of the n-th word being associated with the j-th topic)

$$\boxed{\sum_{d(i)=s} w_{ij} = \sum_{i:\, d(i)=s} w_{ij} = \sum_{i=1}^{l} w_{ij}\, \delta_{d(i),s}} \tag{264}$$

(Probability of the s-th document being associated with the j-th topic)

Well,

$$w_{ij} \overset{(262)}{=} P(y_i = j \mid \{x_i\}, \{\theta\}) = P(y_i = j \mid w = w_{w(i)}, d = d_{d(i)}, \{\theta\}) \overset{(52)}{=}$$

$$= \frac{P(y_i = j, w = w_{w(i)}, d = d_{d(i)} \mid \{\theta\})}{P(w = w_{w(i)}, d = d_{d(i)} \mid \{\theta\})} \overset{(49)}{\overset{(260)}{=}} \frac{\mu_{j,w(i)} \cdot \nu_{j,d(i)} \cdot \lambda_j}{\sum_{j=1}^{K} P(w = w_{w(i)}, d = d_{d(i)} \mid y = j, \{\theta\}) P(y = j \mid \{\theta\})} =$$

$$= \frac{\mu_{j,w(i)} \cdot \nu_{j,d(i)} \cdot \lambda_j}{\sum_{j=1}^{K} P(w = w_{w(i)} \mid y = j, \{\theta\}) P(d = d_{d(i)} \mid y = j, \{\theta\}) \cdot \lambda_j} \overset{(52)}{\overset{(w \perp d \mid y)}{=}} \frac{\mu_{j,w(i)} \cdot \nu_{j,d(i)} \cdot \lambda_j}{\sum_{j=1}^{K} \mu_{j,w(i)} \cdot \nu_{j,d(i)} \cdot \lambda_j} \overset{(167)}{=}$$

$$\boxed{w_{ij} = \frac{\mu_{j,w(i)} \cdot \nu_{j,d(i)} \cdot \lambda_j}{\sum_{j=1}^{K} \mu_{j,w(i)} \cdot \nu_{j,d(i)} \cdot \lambda_j}} \tag{265}$$

Also,

$$P(\{x_i\}, y_i = j \mid \{\theta\}) = P(w = w_{w(i)}, d = d_{d(i)}, y_i = j \mid \{\theta\}) =$$

$$= P(w = w_{w(i)}, d = d_{d(i)} \mid y_i = j, \{\theta\}) \underbrace{P(y_i = j \mid \{\theta\})}_{=\lambda_j} \overset{(52)}{=}$$

$$= P(w = w_{w(i)} \mid y_i = j, \{\theta\}) \cdot P(d = d_{d(i)} \mid y_i = j, \{\theta\}) \cdot \lambda_j \overset{(w \perp d \mid y)}{=}$$

$$\boxed{P(\{x_i\}, y_i = j \mid \{\theta\}) = \mu_{j,w(i)} \cdot \nu_{j,d(i)} \cdot \lambda_j} \tag{266}$$

The "maximization step" of the EM algorithm, assuming the examples $\{x_1\}, \ldots, \{x_l\}$ to be i.i.d. corresponds to (199):

$$\{\theta^{(t+1)}\} = \operatorname*{argmax}_{\{\theta\}} \sum_{j=1}^{K} \sum_{i=1}^{l} w_{ij}^{(t)} \log\left(\lambda_j \cdot \mu_{j,w(i)} \cdot \nu_{j,d(i)}\right) - b\left(\left(\sum_{j=1}^{K} \lambda_j\right) - 1\right)$$
$$- \xi_j\left(\left(\sum_{a=1}^{m} \mu_{ja}\right) - 1\right) - \varepsilon_j\left(\left(\sum_{b=1}^{n} \nu_{jb}\right) - 1\right)$$

subject to $\sum_{j=1}^{K} \lambda_j = 1$, $\sum_{a=1}^{m} \mu_{ja} = 1$, $\sum_{b=1}^{n} \nu_{jb} = 1$ (267)

where λ_i, ξ_j and ξ_j with $j=1,\dots,k$ are Lagrange multipliers.

The Lagrangian of (267) corresponds to

$$\mathcal{L}(\theta,\theta^{(t)}) = \sum_{j=1}^{k}\sum_{i=1}^{l} w_{ij}^{(t)} \log\left(\lambda_j \cdot \mu_{j\,w(i)} \cdot \nu_{j\,d(i)}\right) - \lambda\left(\left(\sum_{j=1}^{k}\lambda_j\right)-1\right) - \xi_j\left(\left(\sum_{a=1}^{m}\mu_{ja}\right)-1\right) - \xi_j\left(\left(\sum_{b=1}^{n}\nu_{jb}\right)-1\right) \tag{268}$$

Well,

$$\frac{\partial\mathcal{L}}{\partial\lambda_r} = \sum_{i=1}^{l}\frac{w_{ir}^{(t)}}{\lambda_r} - \lambda \Leftrightarrow \boxed{\frac{\partial\mathcal{L}}{\partial\lambda_r} = \frac{\left(\sum_{i=1}^{l} w_{ir}^{(t)}\right)-\lambda\lambda_r}{\lambda_r}} \tag{269}$$

$$\left.\frac{\partial\mathcal{L}}{\partial\lambda_r}\right|_{\{\lambda\}=\{\lambda^{(t+1)}\}} = 0 \Leftrightarrow \left(\sum_{i=1}^{l} w_{ir}^{(t)}\right) - \lambda\lambda_r^{(t+1)} = 0 \wedge \lambda_r^{(t+1)} \neq 0 \Leftrightarrow$$

$$\Leftrightarrow \boxed{\lambda_j^{(t+1)} = \frac{\sum_{i=1}^{l} w_{ij}^{(t)}}{\lambda}} \tag{270}$$

By inserting (255) on (270) one gets:

$$\sum_{j=1}^{k}\lambda_j^{(t+1)} = 1 \Leftrightarrow \frac{\sum_{j=1}^{k}\sum_{i=1}^{l} w_{ij}^{(t)}}{\lambda} = 1 \Leftrightarrow \boxed{\lambda = \sum_{j=1}^{k}\sum_{i=1}^{l} w_{ij}^{(t)}} \tag{271}$$

$$\boxed{\lambda_j^{(t+1)} = \frac{\sum_{i=1}^{l} w_{ij}^{(t)}}{\sum_{j=1}^{k}\sum_{i=1}^{l} w_{ij}^{(t)}}} \tag{272}$$

$$\frac{\partial\mathcal{L}}{\partial\mu_{rs}} = \left(\sum_{i=1}^{l}\frac{w_{ir}^{(t)}}{\mu_{rs}}\delta_{w(i)s}\right) - \xi_r = \frac{\left(\sum_{i=1}^{l} w_{ir}^{(t)}\delta_{w(i)s}\right)-\xi_r\mu_{rs}}{\mu_{rs}} \Leftrightarrow$$

$$\left(\sum_{i=1}^{l} w_{ir}^{(t)}\delta_{w(i)s} = \sum_{w(i)=s} w_{ir}^{(t)}\right)$$

$$\Leftrightarrow \boxed{\frac{\partial\mathcal{L}}{\partial\mu_{rs}} = \frac{\left(\sum_{w(i)=s} w_{ir}^{(t)}\right)-\xi_r\mu_{rs}}{\mu_{rs}}} \tag{273}$$

$$\left.\frac{\partial\mathcal{L}}{\partial\mu_{rs}}\right|_{[\mu]=[\mu^{(t+1)}]} = 0 \Leftrightarrow \frac{\left(\sum_{w(i)=s} w_{ir}^{(t)}\right)-\xi_r\mu_{rs}^{(t+1)}}{\mu_{rs}^{(t+1)}} = 0 \Leftrightarrow$$

$$\Leftrightarrow \left(\sum_{w(i)=s} w_{ir}^{(t)}\right)-\xi_r\mu_{rs}^{(t+1)} = 0 \wedge \mu_{rs}^{(t+1)} \neq 0 \Leftrightarrow$$

$$\Leftrightarrow \boxed{\mu_{ja}^{(t+1)} = \frac{\sum_{w(i)=a} w_{ij}^{(t)}}{\xi_j}} \tag{274}$$

By inserting (258) on (274), one gets

$$\boxed{\xi_j = \sum_{a=1}^{m}\sum_{w(i)=a} w_{ij}^{(t)}} \tag{275}$$

$$\boxed{\mu_{ja}^{(t+1)} = \frac{\sum_{w(i)=a} w_{ij}^{(t)}}{\sum_{a=1}^{m}\sum_{w(i)=a} w_{ij}^{(t)}}} \tag{276}$$

Analogously, one may derive

$$\varepsilon_j = \sum_{k=1}^{m} \sum_{d(i)=k} w_{ij}^{(t)} \tag{277}$$

$$v_{jk}^{(t+1)} = \frac{\sum_{d(i)=k} w_{ij}^{(t)}}{\sum_{k=1}^{m} \sum_{d(i)=k} w_{ij}^{(t)}} \tag{278}$$

V- Spectral Analysis I: PCA

Consider the case of sample data $\mathcal{S} = \{x_1, \dots, x_m\}$, with $x_i \in \mathbb{R}^n$. If n and m are too large and $m \gg n$ (as would happen in images with a lot of pixels) it would be beneficial to in some way approximate the sample data to another of lower size - through a so-called dimensionality reduction method. An example of such methods is Principal Component Analysis (PCA).

PCA: Statistical Perspective

Let the sample data $\mathcal{S}$ be centred at $\{0\}$ by making the assignment $\{x_i\} \leftarrow \{x_i\} - \{\mu\}$ with $\{\mu\}$ given by the MLE estimate (93):

$\{\mu\} = \frac{1}{m} \sum_{i=1}^{m} \{x_i\}$, that is

$$\boxed{\{x_i\} \leftarrow \{x_i\} - \frac{1}{m} \sum_{i=1}^{m} \{x_i\} \quad \text{for } i = 1, \dots, m} \tag{279}$$

Note: the since the mean vector estimate $\{\mu\}$ is according to (99) unbiased, the expected value of $\{x_i\}$ given by (279) would be null, that is,

$$\boxed{\mathbb{E}(\{x_i\}) = \{0\}, \text{ for } i = 1, \dots, m} \tag{280}$$

The principle of PCA is finding new variables $y_i^{(1)}, \dots, y_i^{(q)}$ with $q \leq \min(n,m)$ and $i = 1, \dots, m$ from linear combinations of the respective $x_i^{(1)}, \dots, x_i^{(n)}$ variables with $i = 1, \dots, m$, that is,

$$y_i^{(j)} = \sum_{k=1}^{} \underbrace{\mu_{jk}}_{\text{scalars}} x_i^{(k)} = \{\mu_j\}^T \{x_i\} \Rightarrow \begin{Bmatrix} y_i^{(1)} \\ \vdots \\ y_i^{(q)} \end{Bmatrix} = \underbrace{\begin{bmatrix} \{\mu_1\}^T \\ \vdots \\ \{\mu_q\}^T \end{bmatrix}}_{=: [U]^T} \underbrace{\begin{Bmatrix} x_i^{(1)} \\ \vdots \\ x_i^{(m)} \end{Bmatrix}}_{= \{x_i\}} \Rightarrow$$

$$= \{y_i\}$$

$$\Rightarrow \underbrace{\begin{bmatrix} \{y_1\} \cdots \{y_m\} \end{bmatrix}}_{=: [B]} = [U]^T \cdot \underbrace{\begin{bmatrix} \{x_1\} \cdots \{x_m\} \end{bmatrix}}_{=: [A]} \Rightarrow$$

$$\Rightarrow \boxed{[B] = [U]^T [A]} \tag{281}$$

with $\boxed{[A] = \begin{bmatrix} \{x_1\} \cdots \{x_m\} \end{bmatrix}}$ (282), $\boxed{[U] = \begin{bmatrix} \{\mu_1\} \cdots \{\mu_q\} \end{bmatrix}}$ (283) and $\boxed{[B] = \begin{bmatrix} \{y_1\} \cdots \{y_m\} \end{bmatrix}}$ (284),

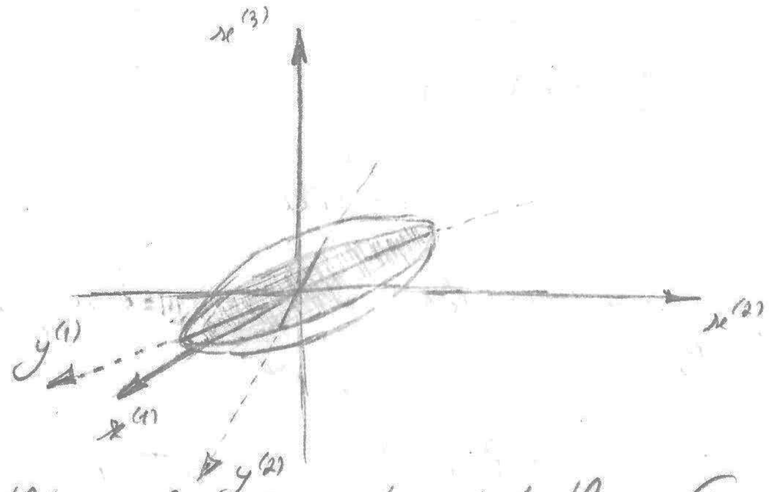
that are decorrelated, that is, with

$$\boxed{\left[\Sigma_{\{y\}}\right]_{jl} = \frac{1}{m} \sum_{i=1}^{m} \left(y_i^{(j)} - \frac{1}{m}\sum_{i=1}^{m} y_i^{(j)}\right)\left(y_i^{(l)} - \frac{1}{m}\sum_{i=1}^{m} y_i^{(l)}\right) = 0 \text{ if } j \neq l} \tag{285}$$

(ML estimator, (96))

and that have maximum variance and decaying in magnitude with the coordinates of $\{y\}$, that is,

$$\boxed{\left[\Sigma_{\{y\}}\right]_{jj} = \max_{[U]} \left[\Sigma_{\{y\}}([U])\right]_{jj}, \text{ with } \left[\Sigma_{\{y\}}\right]_{11} \geq \dots \geq \left[\Sigma_{\{y\}}\right]_{qq}} \tag{286}$$



Note that $\{\mu_j\}$ should be constrained to the condition of $\|\{\mu_j\}\| < \infty$ since $\|\{\mu_j\}\| = \infty$ would trivially maximize $\left[\Sigma_{\{y\}}([U])\right]_{jj}$.

Obtaining the vectors $\{\mu_1\}, \dots, \{\mu_q\}$ can be done through q steps.

First of all one may note that

$$\mathbb{E}(\{y\}) = \frac{1}{m} \sum_{i=1}^{m} \{y_i\} = \frac{1}{m} \sum_{i=1}^{m} [U]^T \{x_i\} = [U]^T \frac{1}{m} \sum_{i=1}^{m} \{x_i\} \Rightarrow$$

(ML estimator, (93)) (281) $= \{0\}$, from (285)

$$\Rightarrow \boxed{\mathbb{E}(\{y\}) = \frac{1}{m} \sum_{i=1}^{m} \{y_i\} = \{0\}}. \tag{287}$$

Let one now consider $j = 1$:

$$\left[\Sigma_{\{y\}}\right]_{11} = \frac{1}{m} \sum_{i=1}^{m} \left(y_i^{(1)} - \underbrace{\frac{1}{m} \sum_{i=1}^{m} y_i^{(1)}}_{=0 \text{ from (287)}}\right)^2 \overset{(281)}{=} \frac{1}{m} \sum_{i=1}^{m} \left(\{\mu_1\}^T \{x_i\}\right)^2 =$$

(ML estimator, (96))

$$= \frac{1}{m} \left[\{\mu_1\}^T \{x_1\} \dots \{\mu_1\}^T \{x_m\}\right] \left[\{\mu_1\}^T \{x_1\} \dots \{\mu_1\}^T \{x_m\}\right]^T =$$

$$= \frac{1}{m} \{\mu_1\}^T \underbrace{\left[\{x_1\} \dots \{x_m\}\right]}_{= [A] \text{ from (282)}} \left(\{\mu_1\}^T [A]\right)^T \Rightarrow$$

$$\Rightarrow \boxed{\left[\Sigma_{yy}\right]_{11} = \frac{1}{m} \{\mu_1\}^T [A][A]^T \{\mu_1\}} \tag{288}$$

Conditions (286) and (288) with $\|\{\mu_j\}\| = \overset{\geq 0}{a} < \infty$ for $j=1$ may be expressed as

$$\{\mu_1^*\} = \underset{\{\mu_1\}}{\operatorname{argmax}} \frac{1}{m} \{\mu_1\}^T [A][A]^T \{\mu_1\} \tag{a}$$

subject to

$$\{\mu_1\}^T \{\mu_1\} = a < \infty$$

$$\Rightarrow \{\mu_1^*\} = \underset{\{\mu_1\}}{\operatorname{argmax}} \frac{a}{m} \frac{1}{\sqrt{a}} \{\mu_1\}^T [A][A]^T \frac{1}{\sqrt{a}} \{\mu_1\}$$

subject to

$$\frac{1}{\sqrt{a}} \{\mu_1\}^T \cdot \frac{1}{\sqrt{a}} \{\mu_1\} = 1 \tag{b}$$

(the maximizer of $\mathcal{L}(\alpha)$ is the same as the one for $c f(\alpha)$ with $c \geq 0$)

$$\Rightarrow \{\mu_1^*\} = \underset{\{\mu_1\}}{\operatorname{argmax}} \frac{1}{\sqrt{a}} \{\mu_1\}^T [A][A]^T \frac{1}{\sqrt{a}} \{\mu_1\}$$

subject to

$$\frac{1}{\sqrt{a}} \{\mu_1\}^T \cdot \frac{1}{\sqrt{a}} \{\mu_1\} = 1$$

(Let $\boxed{\{w_1\} = \frac{\{\mu_1\}}{\sqrt{a}}}$) (289)

$$\Rightarrow \boxed{\{\mu_1^*\} = \sqrt{a}\ \underset{\{w_1\}}{\operatorname{argmax}} \{w_1\}^T [A][A]^T \{w_1\} \quad \text{subject to} \quad \{w_1\}^T \{w_1\} = 1} \tag{290}$$

From (290) and (281) one may conclude that

$$y_i^{(1)} = \sqrt{a}\, \{w_1^*\}^T \{x_i\}$$

where a is to be arbitrated. One may simply arbitrate $a=1$ and re-express (290) as

$$\boxed{\{\mu_1^*\} = \underset{\{\mu_1\}}{\operatorname{argmax}} \{\mu_1\}^T [A][A]^T \{\mu_1\} \quad \text{subject to} \quad \{\mu_1\}^T \{\mu_1\} = 1} \tag{291}$$

The dual form of (291) corresponds to

$$\boxed{\{\mu_1^*\} = \underset{\{\mu_1\}}{\operatorname{argmax}} \mathcal{L}(\{\mu_1\}) \quad \text{subject to} \quad \{\mu_1\}^T \{\mu_1\} = 1} \tag{292}$$

with $\boxed{\mathcal{L}(\{\mu_1\}, \lambda) = \{\mu_1\}^T [A][A]^T \{\mu_1\} - \lambda_1 (\{\mu_1\}^T \{\mu_1\} - 1)}$ where (293)

λ_1 is a Lagrange multiplier. Well,

$$\frac{\partial \mathcal{L}}{\partial \mu_1^{(k)}} = \frac{\partial \{\mu_1\}^T}{\partial \mu_1^{(k)}} [A][A]^T \{\mu_1\} + \{\mu_1\}^T [A][A]^T \frac{\partial \{\mu_1\}}{\partial \mu_1^{(k)}} - \lambda_1 \frac{\partial \{\mu_1\}^T}{\partial \mu_1^{(k)}} \{\mu_1\}$$

$$- \lambda_1 \{\mu_1\}^T \frac{\partial \{\mu_1\}}{\partial \mu_1^{(k)}} =$$

$$= \{e_k\}^T [A][A]^T \{\mu_1\} + \overbrace{\{\mu_1\}^T [A][A]^T \{e_k\}}^{\text{(scalar)}} - \lambda_1 \{e_k\}^T \{\mu_1\}$$

$$- \lambda_1 \underbrace{\{\mu_1\}^T \{e_k\}}_{\text{(scalar)}}$$

$$\Leftrightarrow \boxed{\frac{\partial \mathcal{L}}{\partial \mu_1^{(k)}} = 2\{x_k\}^T([A][A]^T - \lambda_1[I])\{\mu_1\}} \quad (294)$$

$$\left.\frac{\partial \mathcal{L}}{\partial \mu_1^{(k)}}\right|_{\{\mu_1\}=\{\mu_1^*\}} = 0 \ \forall k = 1,\dots,m \Leftrightarrow [I]([A][A]^T - \lambda_1[I])\{\mu_1^*\} = \{0\} \Leftrightarrow$$

$$\Leftrightarrow \boxed{[A][A]^T\{\mu_1^*\} = \lambda_1\{\mu_1^*\}} \quad (295)$$

From condition $\boxed{\{\mu_1^*\}^T\{\mu_1^*\} = 1}$ (296) one gets at (295) that

$$\{\mu_1^*\}^T[A][A]^T\{\mu_1^*\} = \underbrace{\{\mu_1^*\}^T\{\mu_1^*\}}_{=1}\lambda_1 \Leftrightarrow$$

$$\Leftrightarrow \boxed{\lambda_1 = \{\mu_1^*\}^T[A][A]^T\{\mu_1^*\}} \quad (297)$$

From (295) one understands that the optimum $\{\mu\}$ vector that is, $\{\mu_1^*\}$, is an eigenvector of matrix $[A][A]^T$. Also, by inserting $\{\mu_1^*\}$ into (293), one gets

$$\boxed{\mathcal{L}(\{\mu_1^*\},\lambda) = \{\mu_1^*\}^T[A][A]^T\{\mu_1^*\} = \lambda_1} \quad (298)$$

From (298) one understands that from the n eigenvectors of $[A][A]^T$ the one associated with the greatest eigenvalue λ_1 would be the one that maximizes the variance $[\Sigma_{\{y\}}]_{11}$.

Let $\{\mu_1^*\}$ be then such eigenvector being termed first principal axis, and λ_1 be such eigenvalue.

The variable $\boxed{y^{(1)} = \{\mu_1^*\}^T\{x\}}$ (299) is termed the first principal component $\{x\}$.

Let one now compute $\{\mu_2\} \neq \{\mu_1\}$ so that $\boxed{y^{(2)} = \{\mu_2\}^T\{x\}}$ is uncorrelated with $y^{(1)} = \{\mu_1^*\}^T\{x\}$, has maximum (300) variance, and also that $\{\mu_2\}^T\{\mu_2\} = 1$. Well,

$$(285): \ [\Sigma_{\{y\}}]_{12} = \frac{1}{m}\sum_{i=1}^{m}\left(y_i^{(1)} - \underbrace{\frac{1}{m}\sum_{i=1}^{m} y_i^{(1)}}_{=0 \text{ from (287)}}\right)\left(y_i^{(2)} - \underbrace{\frac{1}{m}\sum_{i=1}^{m} y_i^{(2)}}_{=0 \text{ from (287)}}\right) =$$

$$= \frac{1}{m}\sum_{i=1}^{m} y_i^{(1)} y_i^{(2)} = \frac{1}{m}\sum_{i=1}^{m}\{\mu_1^*\}^T\{x_i\}\{\mu_2\}^T\{x_i\} =$$

(299), (300); $\{\mu_2\}^T\{x_i\}$ scalar

$$= \frac{1}{m}\{\mu_1^*\}^T\left(\sum_{i=1}^{m}\{x_i\}\{x_i\}^T\right)\{\mu_2\} = \begin{matrix}\nearrow (\{x_i\} = [A]\{t_i\}) \\ \searrow (\{x_i\}^T = \{t_i\}^T[A]^T)\end{matrix}$$

$$= \frac{1}{m}\{\mu_1^*\}^T\sum_{i=1}^{m}[A]\{t_i\}\{t_i\}^T[A]^T\{\mu_2\} =$$

$$= \frac{1}{m}\{\mu_1^*\}^T[A]\left(\underbrace{\sum_{i=1}^{m}\{t_i\}\{t_i\}^T}_{=[I]}\right)[A]^T\{\mu_2\} =$$

$$= \frac{1}{m}\underbrace{\{\mu_1^*\}^T[A][A]^T\{\mu_2\}}_{\text{scalar}} = \frac{1}{m}\{\mu_2\}^T\underbrace{[A][A]^T\{\mu_1^*\}}_{=\lambda_1\{\mu_1^*\}} \Leftrightarrow$$

$$\Leftrightarrow \quad [\Sigma_{\{y\}}]_{1,2} = \frac{\lambda_1}{m} \underbrace{\{\mu_2\}^T \{\mu_1^*\}}_{\text{scalar}} = 0 \quad \Leftrightarrow \quad \boxed{\{\mu_1^*\}^T \{\mu_2\} = 0 \text{ if } \lambda_1 \neq 0} \qquad (301)$$

($y^{(1)}$ and $y^{(2)}$ are uncorrelated) ($\lambda_1 \geq 0$)

Note: (Matrix $[A][A]^T$ is positive semi-definite)

One may show $[A][A]^T$ corresponds to the covariance matrix $[\Sigma_{\{y\}}]$ scaled by a factor of m:

$$[\Sigma_{\{y\}}] \overset{(86)}{=} \frac{1}{m} \sum_{i=1}^{m} (\{x_i\} - \underbrace{\{\mu\}}_{=0, \text{ from (280)}}) (\{x_i\} - \underbrace{\{\mu\}}_{=0})^T = \frac{1}{m} \sum_{i=1}^{m} \{x_i\}\{x_i\}^T =$$

$$= \frac{1}{m} \sum_{i=1}^{m} [A]\{e_i\}\{e_i\}^T[A]^T = \frac{1}{m}[A]\underbrace{\left(\sum_{i=1}^{m} \{e_i\}\{e_i\}^T\right)}_{=[I]}[A]^T =$$

$$= \frac{1}{m}[A][A]^T \Leftrightarrow \boxed{[A][A]^T = m[\Sigma_{\{y\}}]} \qquad (302)$$

Since $m \geq 0$ and $[\Sigma_{\{y\}}]$ is positive semi-definite, $[A][A]^T$ also is. One would have done the proof in a different way:

$$\{z\}^T[A][A]^T\{z\} = ([A]^T\{z\})^T([A]^T\{z\}) = \{w\}^T\{w\} = \|\{w\}\|^2 \geq 0$$

Note: (ML's estimate for covariance matrix is positive semi-definite)

From (96), one has for $\{z\} \neq \{0\}$,

$$\{z\}^T[\Sigma]\{z\} = \frac{1}{m}\{z\}^T \sum_{i=1}^{m} (\{x_i\} - \{\mu\})(\{x_i\} - \{\mu\})^T \{z\} =$$

$$= \frac{1}{m} \sum_{i=1}^{m} \{z\}^T(\{x_i\} - \{\mu\})\left(\{z\}^T(\{x_i\} - \{\mu\})\right)^T \Leftrightarrow \rightarrow$$

$$\Leftarrow \quad \frac{1}{m} \sum_{i=1}^{m} \underbrace{w_i^2}_{\geq 0} \Leftrightarrow \{z\}^T[\Sigma]\{z\} \geq 0 \qquad \square$$

(Let $w_i = \{z\}^T(\{x_i\} - \{\mu\})$)

The optimization problem corresponds to

$$\{\mu_2^*\} = \underset{\{\mu_2\}}{\text{argmax}}\ \mathcal{L}(\{\mu_2\}, \lambda_2, \gamma_{12}) \quad \text{subject to} \quad \{\mu_2\}^T\{\mu_2\} = 1, \quad \{\mu_1^*\}^T\{\mu_2\} = 0 \text{ if } \lambda_1 \neq 0 \qquad (303)$$

where

$$\mathcal{L}(\{\mu_2\}, \lambda_2, \gamma_{12}) = \{\mu_2\}^T[A][A]^T\{\mu_2\} - \lambda_2(\{\mu_2\}^T\{\mu_2\} - 1) - \begin{cases} \gamma_{12}\{\mu_1^*\}^T\{\mu_2\} & \text{if } \lambda_1 \neq 0 \\ 0 \text{ if } \lambda_1 = 0 \end{cases} \Leftrightarrow$$

$$\Leftrightarrow \boxed{\mathcal{L}(\{\mu_2\}, \lambda_2, \gamma_{12}) = \{\mu_2\}^T[A][A]^T\{\mu_2\} - \lambda_2(\{\mu_2\}^T\{\mu_2\} - 1) - \gamma_{12}\{\mu_1^*\}^T\{\mu_2\}(1 - \delta_{\lambda_1,0})} \qquad (304)$$

(Kronecker delta)

where λ_2 and γ_{12} are Lagrange multipliers.

Well,

$$\frac{\partial \mathcal{L}}{\partial \mu_2(k)} \overset{(304)}{=} 2\{e_k\}^T([A][A]^T - \lambda_2[I])\{\mu_2\} - \gamma_{12}\underbrace{\{\mu_1^*\}^T\{e_k\}}_{\text{scalar}}(1 - \delta_{\lambda_1,0}) \Leftrightarrow$$

$$\Leftrightarrow \boxed{\frac{\partial \mathcal{L}}{\partial \mu_2^{(k)}} = \{\mu_k\}^T \left(2\left([A][A]^T - \lambda_2 [I]\right)\{\mu_2\} - \gamma_{12}\{\mu_1^*\}(1-\delta_{\lambda_1,0}) \right)} \quad (305)$$

$$\left.\frac{\partial \mathcal{L}}{\partial \mu_2^{(k)}}\right|_{\{\mu_2\}=\{\mu_2^*\}} = 0 \;\forall k=1,\dots,n \Leftrightarrow 2\left([A][A]^T - \lambda_2[I]\right)\{\mu_2^*\} - \gamma_{12}\{\mu_1^*\}(1-\delta_{\lambda_1,0}) = 0 \Leftrightarrow$$

$$\Leftrightarrow \boxed{[A][A]^T\{\mu_2^*\} = \lambda_2\{\mu_2^*\} + \frac{1}{2}\gamma_{12}\{\mu_1^*\}(1-\delta_{\lambda_1,0})} \quad (306)$$

By multiplicating (306) by $\{\mu_1^*\}^T$ on the left-hand side, one gets

$$\underbrace{\{\mu_1^*\}^T[A][A]^T\{\mu_2^*\}}_{=\{\mu_2^*\}^T[A][A]^T\{\mu_1^*\}} = \lambda_2\underbrace{\{\mu_1^*\}^T\{\mu_2^*\}}_{=0 \text{ from (296)}} + \frac{1}{2}\gamma_{12}\underbrace{\{\mu_1^*\}^T\{\mu_1^*\}}_{=1, \text{ from (296)}}(1-\delta_{\lambda_1,0})$$

$$= \{\mu_2^*\}^T\underbrace{[A][A]^T\{\mu_1^*\}}_{=\lambda_1\{\mu_1^*\}} = \lambda_1\{\mu_2^*\}^T\{\mu_1^*\} = 0 \text{ from (301)}$$

$$\Leftrightarrow 0 = \lambda_2\{\mu_1^*\}^T\{\mu_2^*\} + \frac{1}{2}\gamma_{12}(1-\delta_{\lambda_1,0}) \Leftrightarrow$$

$$\Leftrightarrow \underbrace{\gamma_{12}(1-\delta_{\lambda_1,0})}_{=0 \text{ if } \lambda_1=0} = \underbrace{-2\lambda_2\{\mu_1^*\}^T\{\mu_2^*\}}_{=0 \text{ if } \lambda_1\neq 0 \text{ from (301)}} \Leftrightarrow$$

$$\Leftrightarrow \left(\gamma_{12}(1-\delta_{\lambda_1,0})=0 \wedge \lambda_2\{\mu_1^*\}^T\{\mu_2^*\}=0 \text{ if } \lambda_1\neq 0\right) \vee$$

$$\left(\gamma_{12}(1-\delta_{\lambda_1,0})=0 \wedge \lambda_2\{\mu_1^*\}^T\{\mu_2^*\}=0 \text{ if } \lambda_1=0\right) \Leftrightarrow$$

$$\Leftrightarrow \begin{cases} \boxed{\gamma_{12}(1-\delta_{\lambda_1,0})=0} \quad (307) \\ \boxed{\lambda_2\{\mu_1^*\}^T\{\mu_2^*\}=0} \quad (308) \end{cases}$$

By inserting (307) into (306), one gets

$$\boxed{[A][A]^T\{\mu_2^*\} = \lambda_2\{\mu_2^*\}} \quad (309)$$

By multiplicating (309) by $\{\mu_2^*\}^T$ on the left-hand side, one gets

$$\{\mu_2^*\}^T[A][A]^T\{\mu_2^*\} = \lambda_2\underbrace{\{\mu_2^*\}^T\{\mu_2^*\}}_{=1 \text{ from normalization condition}} \Leftrightarrow \boxed{\{\mu_2^*\}^T[A][A]^T\{\mu_2^*\} = \lambda_2} \quad (310)$$

From (309) one understands that $\{\mu_2^*\}$ is an eigenvector of $[A][A]^T$. Furthermore, from (310) and (286), one understands that $\{\mu_2^*\}$ is the eigenvector associated with the second greatest eigenvalue, λ_2.

Note also that one may always define $\{\mu_2^*\}$ to be perpendicular to $\{\mu_1^*\}$, that is, with $\boxed{\{\mu_1^*\}^T\{\mu_2^*\}=0}$, (311) since $[A][A]^T$ is a symmetric matrix, and any real symmetric matrix has an orthogonal matrix as diagonalizing matrix (see proof in page 25). In the PCA algorithm (311) is indeed imposed.

From induction one may show that the PCA problem for the $\{u_i\}$ vector corresponds to

$$\boxed{\begin{aligned}&\{u_i^*\} = \underset{\{u_i\}}{\operatorname{argmax}} \ \{u_i\}^T[A][A]^T\{u_i\}\\ &\quad \text{subject to}\\ &\quad \begin{cases}\{u_i\}^T\{u_i\}=1\\ \{u_i\}^T\{u_j\}=0 \text{ for } j\neq i,\ j=1,\dots,q\end{cases}\end{aligned}} \tag{312}$$

so that $\lambda_1 \geq \dots \geq \lambda_q$. Furthermore, the whole PCA problem may be expressed as

$$[U^*] = \underset{[U]}{\operatorname{argmax}} \sum_{i=1}^{q} \{u_i\}^T[A][A]^T\{u_i\}$$

subject to

$$\begin{cases}\{u_i\}^T\{u_i\}=1\\ \{u_i\}^T\{u_j\}=0 \text{ for } i,j=1,\dots,q;\ j\neq i\end{cases} \iff$$

$$\left(\begin{aligned}\sum_{i=1}^{q}\{u_i\}^T[A][A]^T\{u_i\} &= \sum_{i=1}^{q}([U]\{e_i\})^T[A][A]^T[U]\{e_i\} =\\ &= \sum_{i=1}^{q}\{e_i\}^T[U]^T[A][A]^T[U]\{e_i\} =\\ &= \sum_{i=1}^{q}([U]^T[A][A]^T[U])_{ii} =\\ &= \operatorname{trace}([U]^T[A][A]^T[U])\end{aligned}\right)$$

$$\left(\begin{aligned}&\begin{cases}\{u_i\}^T\{u_i\}=1\\ \{u_i\}^T\{u_j\}=0\end{cases} \text{ for } i,j=1,\dots,q;\ j\neq i \iff \{u_i\}^T\{u_j\}=\delta_{ij} \text{ for } i,j=1,\dots,q \iff\\ &\iff ([U]\{e_i\})^T[U]\{e_j\}=\delta_{ij} \text{ for } i,j=1,\dots,q \iff \{e_i\}^T[U]^T[U]\{e_j\}=\delta_{ij} \text{ for } i,j=1,\dots,q\end{aligned}\right) \iff$$

$$\left(\iff ([U]^T[U])_{ij}=\delta_{ij} \text{ for } i,j=1,\dots,q \iff [U]^T[U]=[I]\right)$$

$$\iff \boxed{\begin{aligned}&[U^*] = \underset{[U]}{\operatorname{argmax}} \ \operatorname{trace}([U]^T[A][A]^T[U])\\ &\quad \text{subject to}\\ &\quad \begin{cases}[U]^T[U]=[I]\\ \lambda_1\geq\dots\geq\lambda_q \text{ with } \lambda_i=\{u_i\}^T[A][A]^T\{u_i\}\end{cases}\end{aligned}} \tag{313}$$

(PCA problem)

Note: (If the variable $\{x\}$ follows the multivariate normal distribution (79) and if the components of $\{x\}$ are uncorrelated, then, they are independent)

If $\{x\}$ follows the normal distribution (79) then

$$\boxed{p(\{x\}) = \frac{e^{-\frac{1}{2}(\{x\}-\{\mu_{\{x\}}\})^T[\Sigma_{\{x\}}]^{-1}(\{x\}-\{\mu_{\{x\}}\})}}{\sqrt{(2\pi)^n\,|[\Sigma_{\{x\}}]|}}} \tag{314}$$

Well, being the components of $\{x\}$ uncorrelated, matrix $[\Sigma_{\{x\}}]$ is diagonal, and may be expressed as

$$\boxed{[\Sigma_{\{x\}}] = \sum_{i=1}^{n}\sigma_i^2\{e_i\}\{e_i\}^T} \tag{315}$$

Therefore,

$$\boxed{|[\Sigma_{\{x\}}]| = \prod_{i=1}^{n}\sigma_i^2} \tag{316}$$

$$[\Sigma_{\{x\}}]^{-1}\cdot[\Sigma_{\{x\}}] = [I] \Leftrightarrow \left([\Sigma_{\{x\}}]^{-1}\cdot[\Sigma_{\{x\}}]\right)_{ij} = \delta_{ij} \Leftrightarrow$$

$$\Leftrightarrow \{e_i\}^T\left(\sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{a=1}^{m}[\Sigma_{\{x\}}]^{-1}_{kl}\underbrace{[\Sigma_{\{x\}}]_{la}}_{=\sigma_k^2\delta_{la}\text{ from (315)}}\{e_k\}\{e_a\}^T\right)\{e_j\} = \delta_{ij} \Leftrightarrow$$

$$\Leftrightarrow \sum_{k=1}^{m}\sum_{l=1}^{m}[\Sigma_{\{x\}}]^{-1}_{kl}\,\sigma_k^2\,\underbrace{\{e_i\}^T\{e_k\}}_{=\delta_{ik}}\underbrace{\{e_l\}^T\{e_j\}}_{=\delta_{lj}} = \delta_{ij} \Leftrightarrow$$

$$\Leftrightarrow [\Sigma_{\{x\}}]^{-1}_{ij}\,\sigma_j^2 = \delta_{ij} \Leftrightarrow [\Sigma_{\{x\}}]^{-1}_{ij} = \frac{1}{\sigma_j^2}\delta_{ij} \Leftrightarrow$$

$$\Leftrightarrow \boxed{[\Sigma_{\{x\}}]^{-1} = \sum_{i=1}^{m}\frac{1}{\sigma_i^2}\{e_i\}\{e_i\}^T} \tag{317}$$

By inserting (316) and (317) into (314), one gets

$$f(\{x\}) = \frac{e^{-\frac{1}{2}(\{x\}-\{\mu_{\{x\}}\})^T\left(\sum_{i=1}^{m}\frac{1}{\sigma_i^2}\{e_i\}\{e_i\}^T\right)(\{x\}-\{\mu_{\{x\}}\})}}{\sqrt{(2\pi)^m\prod_{i=1}^{m}\sigma_i^2}} =$$

$$= \frac{e^{-\sum_{i=1}^{m}\frac{1}{\sigma_i^2}(x^{(i)}-\mu^{(i)}_{\{x\}})(x^{(i)}-\mu^{(i)}_{\{x\}})}}{\prod_{i=1}^{m}\sqrt{2\pi\sigma_i^2}} =$$

$$= \prod_{i=1}^{m}\frac{e^{-\frac{1}{2}\frac{(x^{(i)}-\mu^{(i)}_{\{x\}})^2}{\sigma_i^2}}}{\sqrt{2\pi\sigma_i^2}} \Leftrightarrow$$

$$\Leftrightarrow \boxed{f(\{x\}) = \prod_{i=1}^{m} f(x^{(i)})} \tag{318}$$

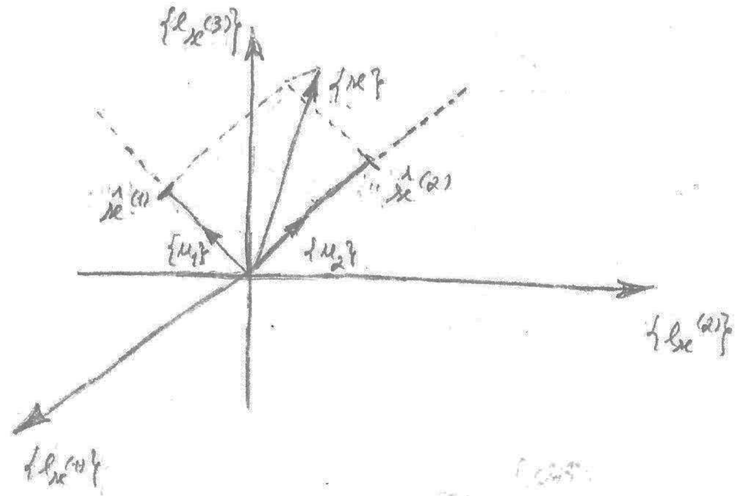
$$\text{with } \boxed{f(x^{(i)}) = \frac{e^{-\frac{1}{2}\left(\frac{x^{(i)}-\mu^{(i)}_{\{x\}}}{\sigma_i}\right)^2}}{\sqrt{2\pi}\,\sigma_i}} \tag{319}$$

From (318) and (319) one concludes that if $\{x\}$ follows a normal distribution and its components $x^{(i)}$ are uncorrelated, then the components $x^{(i)}$ are independent (the PDF for $\{x\}$ is the product of the PDFs for its components $x^{(i)}$).

PCA - Optimal Reconstruction

One may show that the PCA transformation is equivalent to obtain an optimal reconstruction in the least squares sense after dimensional reduction. That is, given a sample data $\{x_1\},\dots,\{x_m\}$, one would like to find orthonormal vectors $\{w_1\},\dots,\{w_q\}$, the so-called principal directions, with $q \le \min(n,m)$, so that the vectors resulting for the projections of the data into these directions – the new variables, $\{\hat{x}_1\},\dots,\{\hat{x}_m\}$ – minimize the residual sum of squares (RSS):

$$\boxed{RSS = \sum_{i=1}^{m}\|\{x_i\}-\{\hat{x}_i\}\|^2} \tag{320}$$



Well, the approximation variable $\{\hat{x}\}$ corresponds to the sum of the projections of $\{x\}$ into $\{u_1\},\dots\{u_q\}$ (with these projections multiplied by the respective vectors $\{u_j\}$):

$$\{\hat{x}\} = \sum_{j=1}^{q} \underbrace{(\{u_j\}^T\{x\})}_{=y^{(j)},\ \text{from (281)}}\{u_j\} = \sum_{j=1}^{q} y^{(j)}\{u_j\} = \sum_{j=1}^{q} \{u_j\} y^{(j)}$$

$$= \sum_{j=1}^{q} [U]\{e_j\}\{e_j\}^T\{y\} = [U]\Big(\underbrace{\sum_{j=1}^{q}\{e_j\}\{e_j\}^T}_{=[I]}\Big)\{y\} \quad (\Leftarrow)$$

$$(\Leftarrow)\quad \boxed{\{\hat{x}\} = [U]\{y\} = [U][U]^T\{x\} = [P]\{x\}} \qquad (321)$$

where $\boxed{\{y\} = [U]^T\{x\}}$, (322) as occurred previously (see (281)), and

$\boxed{[P] = [U][U]^T}$ (322) is the so-called projection matrix.

The optimization problem corresponds to

$$\boxed{\begin{array}{c}[U^*] = \underset{[U]}{\operatorname{argmin}}\ RSS([U]) \\ \text{subject to} \\ [U]^T[U] = [I]\end{array}} \qquad (322)$$

Well

$$RSS([U]) \overset{(320)}{=} \sum_{i=1}^{m} \|\{x_i\} - \{\hat{x}_i\}\|^2 = \sum_{i=1}^{m} (\{x_i\} - \{\hat{x}_i\})^T(\{x_i\} - \{\hat{x}_i\}) =$$

$$\overset{(321)}{=} \sum_{i=1}^{m} (\{x_i\} - [U][U]^T\{x_i\})^T(\{x_i\} - [U][U]^T\{x_i\}) =$$

$$= \sum_{i=1}^{m} \{x_i\}^T([I] - [U][U]^T)([I] - [U][U]^T)\{x_i\} =$$

$$= \sum_{i=1}^{m} \{x_i\}^T([I] - 2[U][U]^T + [U]\underbrace{[U]^T[U]}_{=[I]}[U]^T)\{x_i\} =$$

$$= \Big(\sum_{i=1}^{m} \{x_i\}^T\{x_i\}\Big) - \Big(\sum_{i=1}^{m} \{x_i\}^T[U][U]^T\{x_i\}\Big) =$$

$\{x_i\} = [A]\{e_i\}$

$$= \Big(\sum_{i=1}^{m} \{e_i\}^T[A]^T[A]\{e_i\}\Big) - \Big(\sum_{i=1}^{m} \{e_i\}^T[A]^T[U][U]^T[A]\{e_i\}\Big) =$$

$(\operatorname{trace}([A][B]) = \operatorname{trace}([B][A]))$

$$= \operatorname{trace}([A]^T[A]) - \operatorname{trace}([A]^T[U][U]^T[A]) \quad (\Leftarrow)$$

$$(\Leftarrow)\quad \boxed{RSS([U]) = \operatorname{trace}([A][A]^T) - \operatorname{trace}([U]^T[A][A]^T[U])} \qquad (323)$$

Note: $\left(\text{trace}([A][B]) = \text{trace}([B][A])\right)$

Let us show that

$$\boxed{\text{trace}([A][B]) = \text{trace}([B][A])} \tag{324}$$

Well,

$$\text{trace}([A][B]) = \sum_{i=1} ([A][B])_{ii} = \sum_{i=1} \{u_i\}^T \left(\sum_{j=1} \sum_{l=1} \sum_{k=1} A_{jk} B_{kl} \{u_j\} \{u_l\}^T \right) \{u_i\}$$

$$= \sum_{i=1} \sum_{j=1} \sum_{l=1} \sum_{k=1} A_{jk} B_{kl} \delta_{ji} \delta_{li} = \sum_{i=1} \sum_{k=1} A_{ik} B_{ki}$$

$$\text{trace}([B][A]) = \sum_{i=1} \sum_{k=1} B_{ik} A_{ki} \overset{\text{(change of notation: } i \leftrightarrow k,\ k \leftrightarrow i)}{=} \sum_{k=1} \sum_{i=1} B_{ki} A_{ik} = \sum_{i=1} \sum_{k=1} A_{ik} B_{ki} \tag{3}$$

$$\Rightarrow \text{trace}([B][A]) = \text{trace}([A][B]) \quad \square$$

It would be convenient to order $\{u_j\}$ so that the square of the projection of $\{x\}$ into $\{u_j\}$ decreases (in average) with j:

$$\boxed{\mathbb{E}\left((\hat{x}^{(1)})^2\right) \geq \mathbb{E}\left((\hat{x}^{(2)})^2\right) \geq \ldots \geq \mathbb{E}\left((\hat{x}^{(f)})^2\right)} \tag{325}$$

Well,

$$\mathbb{E}\left((\hat{x}\{u_j\})^2\right) \overset{(321)}{=} \mathbb{E}\left(\left(\{u_j\}^T \{x\}\right)^2\right) \overset{(93)}{=} \frac{\sum_{i=1}^{m} \left(\{u_j\}^T \{x_i\}\right)^2}{m} \quad (\Rightarrow$$

$$\overset{(282)}{\Rightarrow} \mathbb{E}\left((\hat{x}\{u_j\})^2\right) = \frac{\sum_{i=1}^{m} \left(\{u_j\}^T [A] \{x_i\}\right)^2}{m} \overset{\left(\{w\}^T = \{u_j\}^T [A]\right)}{=}$$

$$= \frac{\sum_{i=1}^{m} w_i^2}{m} = \frac{\{w\}^T \{w\}}{m} \quad (\Rightarrow$$

$$\Rightarrow \boxed{\mathbb{E}\left((\hat{x}\{u_j\})^2\right) = \frac{1}{m} \{u_j\}^T [A][A]^T \{u_j\} = \frac{1}{m} \lambda_j} \tag{326}$$

$$\lambda_j$$

where $\boxed{\lambda_j = \{u_j\}^T [A][A]^T \{u_j\}}$ is the eigenvalue of $[A][A]^T$ (327) associated with the eigenvector $\{u_j\}$. Indeed, from (327) one has

$$[A][A]^T \{u_j\} = \lambda_j \{u_j\}$$

One may conclude that the principal direction $\{u_j\}$ with the greatest eigenvalue λ_j is the one whose projection of $\{x\}$ with respect to it is the most close in the least square sense to $\{x\}$.

One may transform (325) into

$$\boxed{\lambda_1 \geq \ldots \geq \lambda_f \text{ with } \lambda_i = \{u_i\}^T [A][A]^T \{u_i\}} \tag{328}$$

By combining (322), (323) and (328) the PCA problem is translated to

$$[U^*] = \underset{[U]}{\text{argmin}} \ \text{trace}([A][A]) - \text{trace}([U]^T [A][A]^T [U]) \quad (\Rightarrow$$

subject to

$$\begin{cases} [U]^T [U] = [I] \\ \lambda_1 \geq \ldots \geq \lambda_f \text{ with } \lambda_i = \{u_i\}^T [A][A]^T \{u_i\} \end{cases}$$

$$\Leftrightarrow \boxed{[U^*] = \underset{[U]}{\operatorname{argmax}} \ \operatorname{trace}\left([U]^T[A][A]^T[U]\right) \quad \text{subject to} \quad [U]^T[U] = [I]}$$

$$\lambda_1 \geq \ldots \geq \lambda_q \text{ with } \lambda_i = \{u_{:i}\}^T[A][A]^T\{u_{:i}\}$$

which is equivalent to (313). (329)

The case $n \gg m$

Matrix $[A][A]^T$ is $n \times n$. Therefore, it has n eigenvectors $\{u_{:i}\}$. If $n \gg m$ such as in the case of a collection of m images of n pixels, then, it would take too much effort to find its eigenvectors and eigenvalues. Fortunately, it may be shown that the number of non-null eigenvalues of $[A][A]^T$ is less or equal than the minimum of the set $\{n, m\}$, that is,

$$\boxed{\left|\{\lambda_i \text{ for } i = 1, \ldots, n : \lambda_i \neq 0\}\right| \leq \min(n, m)} \quad (330)$$

Note: (Proof of (330))

Let one show (330). Matrix $[A]$ is $n \times m$, and, therefore, $[A][A]^T$ is $n \times n$ and $[A]^T[A]$ is $m \times m$. Let one consider the i-th eigenvector of $[A][A]^T$, $\{u_{:i}\}$, and the respective eigenvalue, λ_i. These satisfy

$$\boxed{[A][A]^T\{u_{:i}\} = \lambda_i\{u_{:i}\}} \quad (331)$$

$\leftarrow$

By multiplying (331) by $[A]^T$ on the left, one gets

$$\boxed{[A]^T[A]\underbrace{([A]^T\{u_{:i}\})}_{=:\{v_{:i}\}} = \lambda_i\underbrace{([A]^T\{u_{:i}\})}_{=\{v_{:i}\}}} \quad (332)$$

If $\lambda_i \neq 0$, then $[A]^T\{u_{:i}\} \neq \{0\}$ since $[A]^T\{u_{:i}\} = \{0\}$ would imply from (331) that $[A][A]^T\underbrace{\{u_{:i}\}}_{=\{0\}} = \{0\} \Rightarrow \lambda_i = 0$, as by definition an eigenvector is non-null, that is, $\{u_{:i}\} \neq 0$. Therefore, from (332), one may conclude that any non-null eigenvalue of $[A][A]^T$, λ_i, is also an eigenvalue of $[A]^T[A]$ and the respective eigenvector corresponds to

$$\boxed{\{v_{:i}\} = [A]^T\{u_{:i}\}}. \quad (333)$$

One may conclude that $[A]^T[A]$ and $[A][A]^T$ share the same non-null eigenvalues. Also, since $[A]^T[A]$ has m eigenvalues, one may further say that the number of non-null eigenvalues is less or equal than the minimum of $\{n, m\}$, proving (330).

One may write

$$\boxed{\text{If } [A][A]^T\{u_{:i}\} = \lambda_i\{u_{:i}\} \text{ and } \lambda_i \neq 0, \text{ then } [A]^T[A]([A]^T\{u_{:i}\}) = \lambda_i([A]^T\{u_{:i}\})}. \quad (334)$$

Null eigenvalues λ_j are such that the respective eigenvectors $\{u_j\}$ do not contribute to the transformed feature $\{\hat{x}\}$ and, may be then disregarded. Indeed, from (326),

$$0 = \lambda_j = m\,\mathbb{E}\left((\hat{x}^{(j)})^2\right) = \sum_{i=1}^{m} \underbrace{\left(\hat{x}_i^{(j)}\right)^2}_{\geq 0} \Leftrightarrow \hat{x}_i^{(j)} = 0 \;\forall\; i = 1, \dots, m$$

Therefore instead of diagonalising the very large matrix $[A][A]^T$, one could diagonalise the smaller $[A]^T[A]$ and get $q \leq m$ eigenvectors and respective eigenvalues. However, note that although the non-null eigenvalues of $[A][A]^T$ and $[A]^T[A]$ are the same, the eigenvectors are not. But they are related to each other. Let λ_i and $\{v_i\}$ be the i-th eigenvalue and respective eigenvector of $[A]^T[A]$. They then satisfy

$$[A]^T[A]\{v_i\} = \lambda_i\{v_i\} \Leftrightarrow \boxed{[A][A]^T([A]\{v_i\}) = \lambda_i([A]\{v_i\})} \quad (335)$$

(Multiplying by $[A]$ on the right)

The eigenvectors of $[A][A]^T$ are $[A]\{v_i\}$ subject to normalization, that is

$$\{u_i\} = \frac{[A]\{v_i\}}{\|[A]\{v_i\}\|} = \frac{[A]\{v_i\}}{\sqrt{\{v_i\}^T \underbrace{[A]^T[A]\{v_i\}}_{=\lambda_i\{v_i\}}}} = \frac{1}{\sqrt{\lambda_i}}[A]\frac{\{v_i\}}{\|\{v_i\}\|} \Leftrightarrow$$

$$\Leftrightarrow \boxed{\{u_i\} = \frac{1}{\sqrt{\lambda_i}}[A]\{v_i\}} \quad (336)$$

(The eigenvectors $\{v_i\}$ are chosen to be orthonormal)

(Relation between the orthonormal eigenvectors of $[A][A]^T$, that is, $\{u_i\}$, and the orthonormal eigenvectors of $[A]^T[A]$, that is, $\{v_i\}$, which are associated with the eigenvalues $\lambda_i \neq 0$)

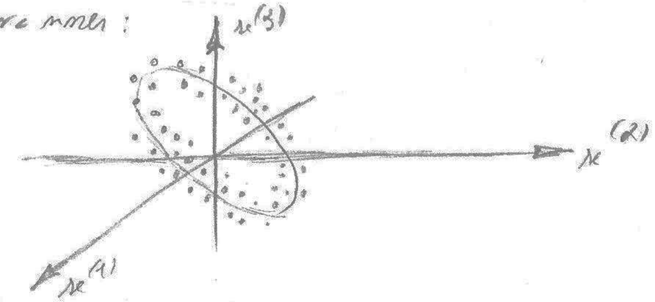
Note that if the eigenvectors $\{v_i\}$ are chosen to be orthogonal then $\{u_i\}$ would also be:

$$\{u_i\}^T\{u_j\} \underset{(336)}{=} \frac{1}{\sqrt{\lambda_i}}\{v_i\}^T[A]^T\frac{1}{\sqrt{\lambda_j}}[A]\{v_j\} = \frac{1}{\sqrt{\lambda_i\lambda_j}}\{v_i\}^T\underbrace{[A]^T[A]\{v_j\}}_{=\lambda_j\{v_j\}} =$$

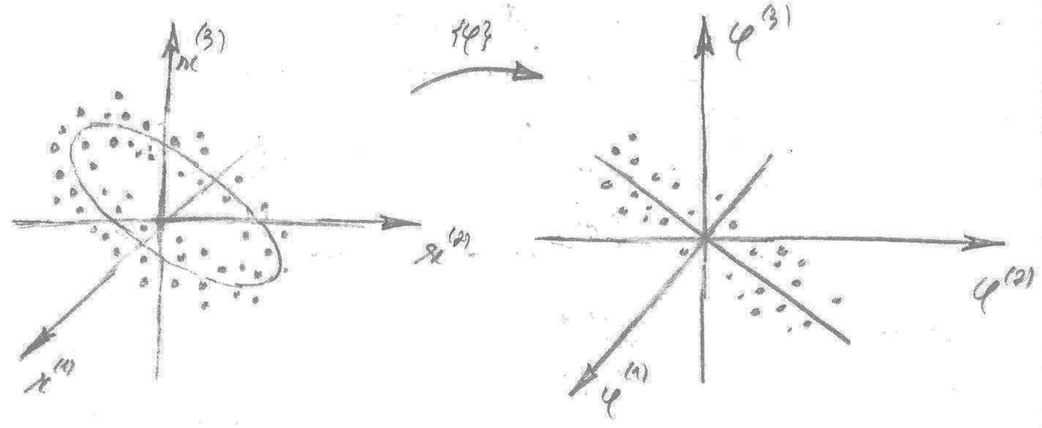
$$= \sqrt{\frac{\lambda_j}{\lambda_i}}\{v_i\}^T\{v_j\} = 0 \text{ if } i \neq j$$

Kernel PCA

Let one consider the case in which the sample data $\{x_1\}, \dots, \{x_m\}$ are distributed in a manner where PCA transformation would produce significant error for q significantly lesser than $\min(n, m)$. Such case could be one in which the data are distributed in an elliptical manner:



PCA would be more effective if one first applied a map $\{\varphi\}: \mathbb{R}^n \to \mathbb{R}^N$ that would transform the ellipsis into a line:



The dimension N might be too big which would make the calculation of each $\{\varphi(\{x_i\})\}$ and the products $\varphi^{(j)}(\{x_i\})\,\varphi^{(k)}(\{x_i\})$ involved in matrices $[A][A]^T$ and $[A]^T[A]$ to be inefficient. Indeed,

$$\left([A][A]^T\right)_{ij} = \sum_k A_{ik} \overbrace{A^T_{kj}}^{=A_{jk}} \quad (=) \quad \left([A][A]^T\right)_{ij} = \sum_k \varphi^{(k)}(\{x_i\})\,\varphi^{(k)}(\{x_j\}) \quad (=)$$

$$\left(A_{ij} = \varphi^{(j)}(\{x_i\})\right)$$

$$(=) \underbrace{\sum_{ij} \left([A][A]^T\right)_{ij} \{e_i\}\{e_j\}^T}_{=[A][A]^T} = \sum_k \underbrace{\left(\sum_i \varphi^{(k)}(\{x_i\})\{e_i\}\right)}_{=\{\varphi(\{x_k\})\}} \cdot \underbrace{\left(\sum_j \varphi^{(k)}(\{x_j\})\{e_j\}^T\right)}_{=\{\varphi(\{x_k\})\}^T} \quad (=)$$

$$(=) \boxed{[A][A]^T = \sum_k \{\varphi(\{x_k\})\}\{\varphi(\{x_k\})\}^T} \qquad (337)$$

$$\left([A]^T[A]\right)_{ij} = \sum_k \overbrace{A^T_{ik}}^{=A_{ki}} A_{kj} = \sum_k \varphi^{(k)}(\{x_i\})\,\varphi^{(k)}(\{x_j\}) =$$

$$= \{\varphi(\{x_i\})\}^T \{\varphi(\{x_j\})\} \quad (=)$$

$$(=) \boxed{[A]^T[A] = \sum_{ij} \{\varphi(\{x_i\})\}^T \{\varphi(\{x_j\})\}\, \{e_i\}\{e_j\}^T} \qquad (338)$$

One way of simplifying the calculations is to use mappings $\{\varphi\}$ which are associated to kernel functions (32):

$$K: \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$$
$$(\{x_i\}, \{x_j\}) \longrightarrow K(\{x_i\},\{x_j\}) = \{\varphi(\{x_i\})\}^T \{\varphi(\{x_j\})\}$$

(Positive-definite kernel function). (339)

However, note that the usage of (339) would not be done in the case of $[A][A]^T$ being employed (see (337)).

By using the kernel K to compute $[A]^T[A]$, the PCA problem would be fully kernelised.

The eigenvectors $\{u_i\}$ and eigenvalues λ_i are obtained from $[A]^T[A]$. The eigenvectors $\{u_i\}$ as defined by (336) cannot be computed in a straightforward manner, though.

But the output vector $\{y\}$ can:

$$(281):\quad \{y\} = [U]^T \{\varphi(\{x\})\} = \sum_{i=1}^{m} \{N_i\}^T \{\varphi(\{x\})\} \{e_i\}$$

$$= \sum_{i=1}^{m} \frac{1}{\sqrt{\lambda_i}} \{N_i\}^T [A]^T \{\varphi(\{x\})\} \{e_i\} =$$

(336)

$$= \sum_{i=1}^{m}\sum_{j=1}^{m} \frac{1}{\sqrt{\lambda_i}} \{N_i\}^T \underbrace{\{\varphi(\{x_j\})\}^T \{\varphi(\{x\})\}}_{= K(\{x_j\},\{x\}),\ \text{from (339)}} \{y\} \{e_i\} =$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m} \frac{1}{\sqrt{\lambda_i}} N_i^{(j)} K(\{x_j\},\{x\}) \{e_i\} =$$

$$= \sum_{i=1}^{m} \frac{\{e_i\}}{\sqrt{\lambda_i}} \underbrace{\sum_{j} N_i^{(j)} K(\{x_j\},\{x\})}_{=[K_{jx}]^T} =$$

$$= \{N_i\}^T \{K\}$$

$$= \left(\sum_{i=1}^{m} \frac{\{e_i\}\{N_i\}^T}{\sqrt{\lambda_i}}\right) \{K\} \quad (=)$$

$$[\Lambda]^{-\frac{1}{2}} [V]^T = \sum_{ijk} [\Lambda]^{-\frac{1}{2}}_{ik} [V]^{\sim^T}_{kj} \{e_i\}\{e_j\}^T$$
$$= \sum_{ijk} \frac{1}{\sqrt{\lambda_i}} \delta_{ik} N_k^{(j)} \{e_i\}\{e_j\}^T$$
$$= \sum_{ij} \frac{\{e_i\}}{\sqrt{\lambda_i}} N_i^{(j)} \{e_j\}^T = \sum_{i} \frac{\{e_i\}\{N_i\}^T}{\sqrt{\lambda_i}}$$

$$(=) \quad \boxed{\{y\} = [\Lambda]^{-\frac{1}{2}} [V]^T \{K\}}, \qquad (340)$$

where $\boxed{[\Lambda] = \sum_{i=1}^{m} \lambda_i \{e_i\}\{e_i\}^T = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{bmatrix}}$, (341)

$\boxed{[V] = \sum_{i=1}^{m} \{N_i\}\{e_i\}^T = [\{N_1\} \cdots \{N_m\}]}$ (342) and

$\boxed{\{K\} = \sum_{i=1}^{m} K(\{x_i\},\{x\}) \{e_i\} = [K(\{x_1\},\{x\}) \cdots K(\{x_m\},\{x\})]^T}$ (343)

Note: $\left([\Lambda]^{-\frac{1}{2}} = \sum_{i=1}^{m} \frac{1}{\sqrt{\lambda_i}} \{e_i\}\{e_i\}^T\right)$

One has

$$[\Lambda]^{-\frac{1}{2}} [\Lambda]^{-\frac{1}{2}} = \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} [\Lambda]^{-1}_{ik} [\Lambda]_{kj} \{e_i\}\{e_j\}^T =$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} \sqrt{\lambda_i}\,\delta_{ik}\,\sqrt{\lambda_k}\,\delta_{kj} \{e_i\}\{e_j\}^T =$$

$$= \sum_{i=1}^{m} \sqrt{\lambda_i}\sqrt{\lambda_i} \{e_i\}\{e_i\}^T = \sum_{i=1}^{m} \lambda_i \{e_i\}\{e_i\}^T =$$

$$= [\Lambda]$$

Therefore one confirms that

$$\boxed{[\Lambda]^{-\frac{1}{2}} = \sum_{i=1}^{m} \frac{1}{\sqrt{\lambda_i}} \{e_i\}\{e_i\}^T}. \qquad (344)$$

Also,

$$[\Lambda]^{-1}[\Lambda] = \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} [\Lambda]^{-1}_{ik} [\Lambda]_{kj} \{e_i\}\{e_j\}^T =$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} \frac{1}{\lambda_i}\,\delta_{ik}\,\lambda_k\,\delta_{kj} \{e_i\}\{e_j\}^T =$$

$$= \sum_{i=1}^{m} \frac{1}{\lambda_i} \lambda_i \{e_i\}\{e_i\}^T = \sum_{i=1}^{m} \{e_i\}\{e_i\}^T = [I].$$

Therefore, one had also confirmed that

$$\boxed{[\Lambda]^{-\frac{1}{2}} = \sum_{i=1}^{m} \frac{1}{\sqrt{\lambda_i}} \{e_i\}\{e_i\}^T} \qquad (345)$$

The approximation vector $\{\varphi(\{\hat{x}\})\}$ may be expressed in a straightforward manner:

$$(321):\ \{\varphi(\{\hat{x}\})\} = [U]\{y\} = [A][V][\Lambda]^{-\frac{1}{2}}\{y\} \overset{(340)}{\Leftarrow}$$

$$\Leftarrow \left((336):\ \{\mu_i\} = [A]\{v_i\}\frac{1}{\sqrt{\lambda_i}} \Leftarrow \boxed{\{\mu_i\} = [A][V][\Lambda]^{-\frac{1}{2}}\{e_i\}}_{(346)} \right)$$

since $[A][V][\Lambda]^{-\frac{1}{2}}\{e_i\} = [A][V]\left(\sum_{j=1}^{m}\frac{1}{\lambda_j}\delta_{ij}\{v_j\}\right) =$

$$= [A]\sum_{j=1}^{m}\frac{\{v_j\}}{\lambda_j}\delta_{ij} = [A]\{v_i\}\frac{1}{\sqrt{\lambda_i}}$$

therefore $[U] = \sum_{i=1}^{m}\{\mu_i\}\{e_i\}^T \overset{(346)}{=} [A][V][\Lambda]^{-\frac{1}{2}}\underbrace{\sum_{i=1}^{m}\{e_i\}\{e_i\}^T}_{=[I]} \Leftarrow$

$$\Leftarrow \boxed{[U] = [A][V][\Lambda]^{-\frac{1}{2}}}_{(347)}$$

$$\Leftarrow \{\varphi(\{\hat{x}\})\} = [A][V][\Lambda]^{-\frac{1}{2}}[\Lambda]^{-\frac{1}{2}}[V]^T\{k\} \Leftarrow$$

$$\Leftarrow \boxed{\{\varphi(\{\hat{x}\})\} = [A][V][\Lambda]^{-1}[V]^T\{k\}}_{(348)}$$

The residual sum of squares RSS is in turn given by

$$RSS = \sum_{i=1}^{m}\left\|\{\varphi(\{x_i\})\} - \{\varphi(\{\hat{x}_i\})\}\right\|^2 =$$

$$= \operatorname{trace}([A][A]^T) = \operatorname{trace}([U]^T[A][A]^T[U]) \overset{(324)}{=}$$

$$\overset{(323)}{=} \underbrace{\operatorname{trace}([A]^T[A])}_{=\operatorname{trace}([K]),\ \text{from (339)}} - \operatorname{trace}\Big(\underbrace{[A]^T[U][U]^T[A]}_{=[B]^T[B],\ \text{from (284)}}\Big) =$$

$$= \operatorname{trace}([K]) - \operatorname{trace}\left(\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m}\underbrace{B^T_{ik}}_{=B_{ki}}B_{kj}\{e_i\}\{e_j\}^T\right) \Leftarrow$$

$$\Leftarrow RSS = \operatorname{trace}([K]) - \operatorname{trace}\left(\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} y_i^{(k)} y_j^{(k)}\{e_i\}\{e_j\}^T\right) =$$

$$\left(B_{ij} = y_j^{(i)}\right)_{(284)}$$

$$= \operatorname{trace}([K]) - \operatorname{trace}\left(\sum_{i=1}^{m}\sum_{j=1}^{m}\{y_i\}^T\{y_j\}\{e_i\}\{e_j\}^T\right) =$$

$$= \operatorname{trace}([K]) - \operatorname{trace}\left(\sum_{i=1}^{m}\{y_i\}^T\{y_i\}\{e_i\}\{e_i\}^T\right) \Leftarrow$$

(trace only takes into account the diagonal entries)

$$\Leftarrow \boxed{RSS = \sum_{i=1}^{m}\left\|\{\varphi(\{x_i\})\} - \{\varphi(\{\hat{x}_i\})\}\right\|^2 = \sum_{i=1}^{m}\left(K_{ii} - \{y_i\}^T\{y_i\}\right)},$$

$$(349)$$

where $[K]$ is the Gram matrix with entries

$$\boxed{K_{ij} = \{\varphi(\{x_i\})\}^T\{\varphi(\{x_j\})\}} \quad (350)$$

and $\{y_i\}$ is given by (340) with $\{k\} = \{K_i\} \overset{(343)}{=} \sum_{j=1}^{m}K_{ji}\{e_j\} \Leftarrow$

$$\Leftarrow \boxed{\{K_i\} = \sum_{j=1}^{m}K_{ij}\{e_j\}}. \quad (351)$$

($[K]$ is symmetric)

Note: (Etymology of "Spectral Analysis")

The designation "Spectral Analysis" was coined by David Hilbert around 1900 - 1910 when he used it in his works involving eigendecompositions. It was actually a fortunate coincidence that it was later used by Heisenberg and other physicists that inferred that the spectral lines seen with prisms or gratings were related to eigenvalues of certain linear operators in Quantum Mechanics.

Table of Contents

VI - Spatial Analysis II: Clustering ——— 191

VII - Regression ——— 209

VIII - K-Nearest Neighbours ——— 233

IX - Decision Trees ——— 239

X - Logistic Regression ——— 255

XI - Bias-Variance Tradeoff ——— 269

XII - Model Performance Assessment ——— 295

XIII - Neural Networks ——— 311

XIV - Gradient Descent ——— 331

XV - Normalization ——— 353

VI- Spectral Analysis II : Clustering

K-means Algorithm (or Lloyd's Algorithm) for Clustering

Let we consider a sample data $\mathcal{D} = \{x_1, \dots, x_M\}$, to be clustered into K groups. Let $y_i \in \{1, \dots, K\}$ be the cluster number (one of the K ones) to be assigned to the i-th data point. The K-means algorithm is such that for given centre vectors of the clusters $\{c_1, \dots, c_K\}$, the cluster to assign to the point x_i corresponds to the one whose centre vector is the closest one to the point, that is,

$$y_i = \arg\min_j \|x_i - c_j\|^2 \tag{351}$$

Note that above, the term "given centre vectors" was used because the algorithm is actually interative and requires initial estimates for the centre vectors $\{c_1, \dots, c_K\}$. Several initialization methods will be introduced later. For now, let's finalize the description of the algorithm steps. Let t refer to the t-th iteration. Eq (351) is termed "assignment step" and may be rewritten as

$$\boxed{y_i^{(t)} = \underset{j}{\operatorname{argmin}} \left\| \{x_i\} - \{c_j^{(t)}\} \right\|^2} \tag{352}$$

("Assignment step" for the i-th point: a cluster number $y_i^{(t)} \in \{1, \dots, k\}$ is assigned to the i-th data point $\{x_i\}$, corresponding to the number of the cluster that is the closest one to the point)

After the clusters being assigned, it is necessary to update the clusters' centre vectors $\{c_1\}, \dots, \{c_k\}$. These are actually mean vectors, that is, they correspond to the arithmetic means of the data points $\{x_i\}$ that pertain to the respective clusters:

$$\boxed{\{c_j^{(t+1)}\} = \frac{1}{\left| \mathcal{S}_j^{(t)} \right|} \sum_{i \in \mathcal{S}_j^{(t)}} \{x_i\}} \tag{353}$$

("Update step" for the j-th cluster: the mean vector of the j-th cluster, $\{c_j\}$ is updated with the arithmetic mean of the data points $\{x_i\}$ that pertain to it)

where $\boxed{\mathcal{S}_j^{(t)} = \{ i : y_i^{(t)} = j \;\; \forall i = 1, \dots, m \}}$ (354) is the set of indices of the data points $\{x_i\}$ that are associated with the j-th cluster at the t-th iteration.

Note that one may also show that $\{c_j^{(t+1)}\}$ is also the point whose average of the distances from the data points pertaining to the j-th cluster at the t-th iteration is the minimum one:

$$\underset{\{c\}}{\operatorname{argmin}} \frac{1}{\left| \mathcal{S}_j^{(t)} \right|} \sum_{i \in \mathcal{S}_j^{(t)}} \left\| \{x_i\} - \{c\} \right\|^2 =$$

$$= \underset{\{c\}}{\operatorname{argmin}} \sum_{i \in \mathcal{S}_j^{(t)}} \left\| \{x_i\} - \{c\} \right\|^2 = \underset{\{c\}}{\operatorname{argmin}} \sum_{i \in \mathcal{S}_j^{(t)}} (\{x_i\} - \{c\})^T (\{x_i\} - \{c\}) =$$

$$= \underset{\{c\}}{\operatorname{argmin}} \sum_{i \in \mathcal{S}_j^{(t)}} \sum_{l=1}^{m} \left(x_i^{(l)} \right)^2 - 2 \left(\sum_{l=1}^{m} c^{(l)} x_i^{(l)} \right) + \sum_{l=1}^{m} \left(c^{(l)} \right)^2 =$$

$$\left(\{x_i\} \in \mathbb{R}^m, \; \{c\} \in \mathbb{R}^m \right)$$

$$= \underset{\{c\}}{\operatorname{argmin}} \sum_{i \in \mathcal{S}_j^{(t)}} \sum_{l=1}^{m} c^{(l)} \left(c^{(l)} - 2 x_i^{(l)} \right) = \frac{1}{\left| \mathcal{S}_j^{(t)} \right|} \sum_{i \in \mathcal{S}_j^{(t)}} \{x_i\}$$

$$\left(\frac{\partial}{\partial c^{(a)}} \left(\sum_{i \in \mathcal{S}_j^{(t)}} \sum_{l=1}^{m} c^{(l)} \left(c^{(l)} - 2 x_i^{(l)} \right) \right) = \sum_{i \in \mathcal{S}_j^{(t)}} c^{(a)} - 2 x_i^{(a)} + c^{(a)} \right.$$

$$= 2 \left(\left| \mathcal{S}_j^{(t)} \right| c^{(a)} - \sum_{i \in \mathcal{S}_j^{(t)}} x_i^{(a)} \right)$$

$$\left. \frac{\partial}{\partial c^{(a)}} \left(\sum_{i \in \mathcal{S}_j^{(t)}} \sum_{l=1}^{m} c^{(l)} \left(c^{(l)} - 2 x_i^{(l)} \right) \right) = 0 \Leftrightarrow c^{(a)} = \frac{1}{\left| \mathcal{S}_j^{(t)} \right|} \sum_{i \in \mathcal{S}_j^{(t)}} x_i^{(a)} \right)$$

Therefore,

$$\boxed{\{c_j^{(t+1)}\} = \underset{\{c\}}{\operatorname{argmin}} \sum_{i \in \mathcal{S}_j^{(t)}} \left\| \{x_i\} - \{c\} \right\|^2} \tag{355}$$

From (352) and and (354), one may understand that the K-means problem may be summarized as the minimization of an overall cost function:

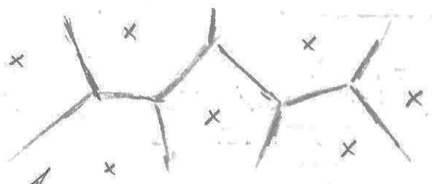
$$\min_{y_1,\dots,y_m,\{c_1\},\dots,\{c_k\}} \sum_{j=1}^{k} \sum_{x\in S_j} \|\{x_i\}-\{c_j\}\|^2 \tag{356}$$

(K-means problem)

Let $\mathcal{C}(y_1,\dots,y_m,\{c_1\},\dots,\{c_k\}) = \sum_{j=1}^{k} \sum_{i\in S_j} \|\{x_i\}-\{c_j\}\|^2$ (357)

be the cost function in (356). One may understand that step (352) minimizes each term (loss function) in (357), and step (353) minimizes each combination of terms associated with each S_j. The two steps then minimize $\mathcal{C}$ and the algorithm necessarily converges, which happens when there is no change in one of the steps.

Also, it is to understand that a point $\{x_i\}$ pertains to the j-th cluster if it is within its Voronoi cell:



Voronoi cell associated with centroid

Matrix Formulation of K-means

Let one expand $\mathcal{C}(y_1,\dots,y_m,\{c_1\},\dots,\{c_k\})$:

$$\mathcal{C} = \sum_{j=1}^{k} \sum_{i\in S_j} \|\{x_i\}-\{c_j\}\|^2 = \sum_{j=1}^{k} \sum_{i\in S_j} (\{x_i\}-\{c_j\})^T(\{x_i\}-\{c_j\}) =$$

$$= \sum_{j=1}^{k} \sum_{i\in S_j} \{x_i\}^T\{x_i\} - 2\{c_j\}^T\{x_i\} + \{c_j\}^T\{c_j\} =$$

$$\left(\text{(353): } \{c_j\} = \frac{1}{|S_j|}\sum_{x\in S_j}\{x_n\}\right)$$

$$= \left(\sum_{j=1}^{k}\sum_{i\in S_j}\{x_i\}^T\{x_i\}\right) - 2\left(\sum_{j=1}^{k}\sum_{i\in S_j}\frac{1}{|S_j|}\sum_{x\in S_j}\{x_n\}^T\{x_i\}\right) +$$

$$+ \sum_{j=1}^{k}\left(\underbrace{\sum_{i\in S_j} 1}_{=|S_j|}\right)\frac{1}{|S_j|}\sum_{\mu\in S_j}\{x_\mu\}^T\frac{1}{|S_j|}\sum_{N\in S_j}\{x_N\}\Bigg) =$$

$$= \left(\sum_{j=1}^{k}\sum_{i\in S_j}\{x_i\}^T\{x_i\}\right) - 2\left(\sum_{j=1}^{k}\frac{1}{|S_j|}\sum_{i\in S_j}\sum_{n\in S_j}\{x_n\}^T\{x_i\}\right) (=$$

$$+ \left(\sum_{j=1}^{k}\frac{1}{|S_j|}\sum_{\mu\in S_j}\sum_{N\in S_j}\{x_\mu\}^T\{x_N\}\right) (=)$$

$$(=) \quad \boxed{\mathcal{C} = \left(\sum_{i=1}^{m}\{x_i\}^T\{x_i\}\right) - \left(\sum_{j=1}^{k}\frac{1}{|S_j|}\sum_{i\in S_j}\sum_{n\in S_j}\{x_n\}^T\{x_i\}\right)} \tag{358}$$

Therefore, since the first term does not depend on $y_1, \dots, y_m$ it may be disregarded and (356) may be transformed into

$$\max_{y_1, \dots, y_m} \sum_{j=1}^{K} \frac{1}{|\mathcal{C}_j|} \sum_{i \in \mathcal{C}_j} \sum_{n \in \mathcal{C}_j} \{x_n\}^T \{x_i\} \tag{359}$$

From (359), one understands that the problem may be kernelized since the dependence on $\{x_i\}$ is in the form of $\{x_n\}^T \{x_i\}$, allowing the mapping

$$(\{x_n\}, \{x_i\}) \longrightarrow K(\{x_n\}, \{x_i\}) = \{\varphi(\{x_n\})\}^T \{\varphi(\{x_i\})\}$$

Let indeed substitute $\{x_n\}^T \{x_i\}$ in (359) by $K_{ni} = K(\{x_n\}, \{x_i\})$: (360)

$$\max_{y_1, \dots, y_m} \sum_{j=1}^{K} \frac{1}{|\mathcal{C}_j|} \sum_{i \in \mathcal{C}_j} \sum_{n \in \mathcal{C}_j} K_{ni} \quad (=)$$

Let $F_{nN}^{(j)} = \begin{cases} \frac{1}{|\mathcal{C}_j|} & \text{if } n, N \in \mathcal{C}_j \\ 0, & \text{if otherwise} \end{cases}$ (361)

By sorting points $\{x_i\}$ according to cluster membership, then one may define a matrix $[F]$ with the matrices $[F^{(1)}], \dots, [F^{(K)}]$ in the diagonal as blocks, where

$$F_{nN}^{(j)} = \frac{1}{|\mathcal{C}_j|}, \quad [F^{(j)}] = \frac{1}{|\mathcal{C}_j|} \{1\}\{1\}^T \tag{362}$$

($\{1\}$: vector of $|\mathcal{C}_j|$ ones; $|\mathcal{C}_j| \times |\mathcal{C}_j|$ matrix of ones)

The rest of the entries of $[F]$ are nulls.

Notice $[F]$ then looks like

$$[F] = \begin{bmatrix} [F_1] & & [0] \\ & \ddots & \\ [0] & & [F_K] \end{bmatrix} = \begin{bmatrix} \frac{1}{|\mathcal{C}_1|} \cdots \frac{1}{|\mathcal{C}_1|} & 0 \cdots 0 \\ \vdots \quad \ddots \quad \vdots & \vdots \quad \ddots \quad \vdots \\ \frac{1}{|\mathcal{C}_1|} \cdots \frac{1}{|\mathcal{C}_1|} & 0 \cdots 0 \\ 0 \cdots 0 & \ddots \\ & \frac{1}{|\mathcal{C}_K|} \cdots \frac{1}{|\mathcal{C}_K|} \\ & \vdots \quad \ddots \quad \vdots \\ 0 \cdots 0 & \frac{1}{|\mathcal{C}_K|} \cdots \frac{1}{|\mathcal{C}_K|} \end{bmatrix}$$

$$\max_{y_1, \dots, y_m} \sum_{j=1}^{K} \frac{1}{|\mathcal{C}_j|} \sum_{i \in \mathcal{C}_j} \sum_{n \in \mathcal{C}_j} K_{ni} = \max_{[F]} \sum_{i=1}^{m} \sum_{j=1}^{m} K_{ij} F_{ij} =$$

$$= \max_{[F]} \sum_{i=1}^{m} \sum_{j=1}^{m} K_{ij} F_{ji} = \max_{[F]} \sum_{i=1}^{m} \{e_i\}^T [K][F]\{e_i\} =$$

([F] is symmetric)

$$= \max_{[F]} \sum_{i=1}^{m} ([K][F])_{ii}$$

The K-means problem then becomes

$$\max_{[F]} \operatorname{trace}([K][F]) \tag{363}$$

Now, let $G_{ij} = \begin{cases} \frac{1}{\sqrt{|\mathcal{C}_j|}}, & \text{if } i \in \mathcal{C}_j \\ 0, & \text{if otherwise} \end{cases}$. (364)

One may show that $\boxed{[F] = [G][G]^T}$ (365)

Indeed,

$$([G][G]^T)_{ij} = \sum_{l=1}^{m} G_{il} G^T_{lj} = \sum_{l=1}^{m} G_{il} G_{jl} = \overset{(364)}{\Leftarrow}$$

$$= \sum_{l=1}^{m} \left(\begin{cases} \frac{1}{\sqrt{|\mathcal{S}_l|}}, & \text{if } i \in \mathcal{S}_l \\ 0, & \text{if otherwise} \end{cases} \right) \cdot \left(\begin{cases} \frac{1}{\sqrt{|\mathcal{S}_l|}}, & \text{if } j \in \mathcal{S}_l \\ 0, & \text{if otherwise} \end{cases} \right) =$$

$$= \sum_{l=1}^{m} \left(\begin{cases} \frac{1}{|\mathcal{S}_l|} & \text{if } i,j \in \mathcal{S}_l \\ 0, & \text{if otherwise} \end{cases} \right) = \sum_{l=1}^{m} \frac{1}{|\mathcal{S}_l|} \delta_{y_i l} \delta_{y_j l} =$$

$$= \sum_{l=1}^{m} \frac{1}{|\mathcal{S}_l|} \delta_{y_i l} \delta_{y_j y_i} \Leftarrow \boxed{([G][G]^T)_{ij} = \frac{1}{|\mathcal{S}_{y_i}|} \delta_{y_i y_j} = F_{ij}} \quad (366)$$

Problem (363) may be then transformed into

$$\min_{[G]} \operatorname{trace}([K][G][G]^T) = \min_{[G]} \operatorname{trace}(([K][G])[G]^T) =$$

$$\overset{(304)}{\nearrow} = \boxed{\min_{[G]} \operatorname{trace}([G]^T[K][G])} \quad (367)$$

In (367), it is implicit that $[G]$ satisfy (364). Let one now infer properties of $[G]$.

From (364), one has

$$\boxed{G_{ij} \geq 0} \quad (368)$$

Also,

$$([G]^T[G])_{ij} = \sum_{l=1}^{m} G^T_{il} G_{lj} = \sum_{l=1}^{m} G_{li} G_{lj} = \overset{(364)}{\Leftarrow}$$

$$= \sum_{l=1}^{m} \left(\begin{cases} \frac{1}{\sqrt{|\mathcal{S}_i|}}, & \text{if } l \in \mathcal{S}_i \\ 0, & \text{if otherwise} \end{cases} \right) \left(\begin{cases} \frac{1}{\sqrt{|\mathcal{S}_j|}}, & \text{if } l \in \mathcal{S}_j \\ 0, & \text{if otherwise} \end{cases} \right) =$$

$$= \sum_{l=1}^{m} \begin{cases} \frac{1}{\sqrt{|\mathcal{S}_i|}\sqrt{|\mathcal{S}_j|}}, & \text{if } l \in \mathcal{S}_i \wedge l \in \mathcal{S}_j \\ 0, & \text{if otherwise} \end{cases} =$$

$$= \sum_{l=1}^{m} \frac{1}{\sqrt{|\mathcal{S}_i|}\sqrt{|\mathcal{S}_j|}} \delta_{y_l i} \delta_{y_l j} =$$

$$= \sum_{l=1}^{m} \frac{1}{|\mathcal{S}_i|} \cdot \delta_{y_l i} \delta_{ij} =$$

$$= \left(\sum_{l \in \mathcal{S}_i}^{n} \frac{1}{|\mathcal{S}_i|} \right) \delta_{ij} = \underbrace{\frac{|\mathcal{S}_i|}{|\mathcal{S}_i|}}_{=1} \delta_{ij} = \delta_{ij} \Leftarrow$$

$$\Leftarrow \boxed{[G]^T[G] = [I]} \quad (369)$$

Matrix $[G][G]^T$ is a right stochastic matrix, that is, the sums of each row are 1:

$$\sum_{j=1}^{nr} ([G][G]^T)_{ij} = \sum_{j=1}^{nr} \frac{1}{|\mathcal{S}_{y_i}|}\,\delta_{y_i,y_j} = \sum_{j\in\mathcal{S}_{y_i}} \frac{1}{|\mathcal{S}_{y_i}|} = \frac{|\mathcal{S}_{y_i}|}{|\mathcal{S}_{y_i}|} \quad (366)$$

$$\Leftrightarrow \sum_{j=1}^{nr} ([G][G]^T)_{ij} = 1 \quad \forall\, i=1,\dots,nr \Leftrightarrow$$

$$\Leftrightarrow \{1_i\}^T [G][G]^T \{1\} = 1 \;\forall\, i=1,\dots,nr \Leftrightarrow [I][G][G]^T\{1\} = \{1\} \Leftrightarrow$$

$$\Leftrightarrow \boxed{[G][G]^T\{1\} = \{1\}} \quad (370)$$

And since $[G][G]^T$ is a symmetric matrix, it is also left stochastic matrix, that is, the sums of each column are 1:

$$\sum_{i=1}^{nr} ([G][G]^T)_{ij} = \sum_{i=1}^{nr} ([G][G]^T)^T_{ji} = \sum_{i=1}^{nr} ([G][G]^T)_{ji} = \sum_{j=1}^{nr} ([G][G]^T)_{ij} \overset{(370)}{\Rightarrow}$$

$$\Leftrightarrow \sum_{i=1}^{nr} ([G][G]^T)_{ij} = 1 \quad \forall\, j=1,\dots,nr. \Leftrightarrow$$

$$\Leftrightarrow \{1\}^T([G][G]^T)\{e_j\} = 1 \;\forall\, j=1,\dots,nr \Leftrightarrow \{1\}^T[G][G]^T[I] = \{1\}^T \Leftrightarrow$$

$$\Leftrightarrow \boxed{\{1\}^T[G][G]^T = \{1\}^T} \quad (371)$$

Matrix $[G][G]^T$ is said to be doubly stochastic since it is both right and left stochastic.

One may show that constraints (368), (369) and (370) ((371) follows from (370) and the fact that any matrix $[G][G]^T$ is symmetric) are equivalent to definition (364). Indeed, from (369), one has

$$([G]^T[G])_{ij} = 0,\ \text{if } i\neq j \Leftrightarrow \sum_{l=1}^{nr} G^T_{il} G_{lj} = 0,\ \text{if } i\neq j \Leftrightarrow \sum_{l=1}^{nr} G_{li} G_{lj} = 0 \text{ if } i\neq j$$

$\Rightarrow$ If an entry of the i-th column of $[G]$ is non-null, the respective entry of the j-th column is null, for any $i\neq j$ $\Leftrightarrow$

(368: $G_{li}\geq 0$, $G_{lj}\geq 0$)

$\Rightarrow$ $\boxed{\text{Each row of } [G] \text{ has at most a non-null entry}}$ (372)

(Implication of $G_{ij}\geq 0\ \forall\, i,j=1,\dots,nr$ and $([G]^T[G])_{ij}=0\ \forall\, i,j=1,\dots,nr$ if $i\neq j$)

From (370) one has

$$\sum_{j=1}^{nr} ([G][G]^T)_{ij} = 1 \Leftrightarrow \sum_{j=1}^{nr}\sum_{l=1}^{nr} G_{il}\, G^T_{lj} = 1 \Leftrightarrow \boxed{\sum_{j=1}^{nr}\sum_{l=1}^{nr} G_{il}\, G_{jl} = 1 \;\forall\, i=1,\dots,nr} \quad (373)$$

From (373), one understands that there is a least one entry of each row that is non-null. If all were null, then, $\sum_{l=1}^{nr} G_{il} G_{jl} = \sum_{l=1}^{nr} 0 \cdot G_{jl} = 0$. Therefore, from (372) and (373), one concludes that each row of $[G]$ has one and only one non-null entry.

Only terms $G_{il}G_{jl}$ that are non-null contribute to the sum in (373). There are such that $G_{il}>0$, $G_{jl}>0$, that is, both i and j-th entries of the l-th column of $[G]$ need to be non-null. Let $\{u_l\}$ be the vector of non-null entries of the l-th column. Expression (373) may be transformed into

$$\sum_{j=1}^{m}\sum_{l=1}^{m} G_{il}G_{jl}=1 \Leftrightarrow G_{if}\left(\sum_{j=1} G_{jf}\right)=1 \Leftrightarrow$$

(f be the index of the non-null entry of the i-th row)

($|\mathcal{S}_l|$ is the number of entries of $\{u_l\}$)

($q=1:\ u_l^{(1)}=G_{if}$; $s=a:\ u_f^{(a)}=G_{jf}$)

$$\Leftrightarrow \boxed{\mu_f^{(q)}\left(\sum_{s=1}^{|\mathcal{S}_f|}\mu_f^{(s)}\right)=1 \quad \forall f=1,\dots,m,\ q=1,\dots,|\mathcal{S}_f|} \tag{374}$$

By summing (374) on $q=1,\dots,|\mathcal{S}_f|$ one gets

$$\sum_{q=1}^{|\mathcal{S}_f|}\mu_f^{(q)}\left(\sum_{s=1}^{|\mathcal{S}_f|}\mu_f^{(s)}\right)=\left(\sum_{q=1}^{|\mathcal{S}_f|}1\right)\Leftrightarrow\left(\sum_{s=1}^{|\mathcal{S}_f|}\mu_f^{(s)}\right)^2=|\mathcal{S}_f|\Leftrightarrow$$

($\sum_{q=1}^{|\mathcal{S}_f|}1=|\mathcal{S}_f|$)

$$\Leftrightarrow \boxed{\sum_{s=1}^{|\mathcal{S}_f|}\mu_f^{(s)}=\sqrt{|\mathcal{S}_f|}} \tag{375}$$

And by inserting (375) into (374) one obtains

$$\boxed{\mu_f^{(q)}=\frac{1}{\sqrt{|\mathcal{S}_f|}}} \tag{376}$$

Therefore, (368), (369) and (370) is equivalent to definition (364). Hard problem (367) may be transformed into

$$\boxed{\begin{array}{ll}\text{minimize} & \text{trace}\left([G]^T[k][G]\right)\\ [G] & \text{subject to}\\ & \begin{cases}G_{ij}\geq 0 \quad \forall i,j=1,\dots,m\\ [G]^T[G]=[I]\\ [G][G]^T\{1\}=\{1\}\end{cases}\end{array}} \tag{377}$$

(K-means problem)

Problem (377) is actually equivalent to the PCA problem (313) in which the eigendecomposition is on matrix $[A]^T[A]$ and not $[A][A]^T$ with $[A]^T[A]$ substituted by the Gram matrix $[K]$ and subject to two extra constraints: $G_{ij}\geq 0$ and $[G][G]^T\{1\}=\{1\}$.

Forgy initialisation

Forgy initialisation corresponds to randomly define K centre vectors as K points of the sample data.

Random partition initialisation (not recommended)

This method is such that a cluster is randomly assigned to each point and from them the centre vectors are computed. Such method is not recommended since it usually produces centre vectors close to each other and to the centre of the whole sample set. Indeed the expected value for a point of a cluster defined in this way is the centre of the whole sample data:

$$P(\{x_i\} \mid y_i = j) = P(\{x_i\}) = \frac{1}{m}$$

($\{x\}$ and y are independent) (the points are randomly picked with uniform probability)

$$\mathbb{E}(\{x_i\} \mid y_i = j) = \sum_{i=1}^{m} \{x_i\} P(\{x_i\} \mid y_i = j) = \frac{1}{m} \sum_{i=1}^{m} \{x_i\} = \{c_0\}$$

K-means ++ initialisation (recommended)

The K-means ++ initialisation allows one to define K points as centre vectors that are probabilistically the furthest from each other. It consists on

- Randomly picking one point from the sample set and choosing it to be the first centre vector, $\{c_1\}$
- Repeating the following for $i = 2, \dots, K$:
  - Defining the probability for choosing the l-th point to be the i-th centre vector, $\{c_i\}$, as being proportional to the smallest of the distances between the point $\{x_l\}$ and the already defined centre vectors, $\{c_1\}, \dots, \{c_{i-1}\}$:

$$P(\{c_i\} = \{x_l\}) = \frac{D(\{x_l\})}{\sum_{l=1}^{m} D(\{x_l\})}$$

where (378)

$$D(\{x_l\}) = \min\left(\left\{\|\{x_l\} - \{c_j\}\|^2\right\}_{j=1,\dots,i-1}\right) \quad (379)$$

  - Randomly choosing a point to be the i-th centre vector, according to probability (378).

Note that points associated with the previous centre vectors are never chosen again, as these have null probability

The K-means++ initialization algorithm is the recommended one.

VII – Regression

Regression is a Supervised Learning method that estimates the relationship between Feature $\{x\}$ and a label y. This label is in turn a continuous variable. The learning is said to be supervised since the training data corresponds to Features $\{x\}$ that were already labelled. Regression differs from Classification in what concerns the admissible values of the label y: in Classification, y is discrete and limited.

Linear Regression

Let $\{x\} \in \mathbb{R}^D$ be the features and $y \in \mathbb{R}$ the label. In Linear Regression the relationship between y and $\{x\}$ is assumed to be linear, that is,

$$\boxed{y = \{w\}^T \{x\} + b} \tag{380}$$

where $\{w\}$ is a constant weight vector, and b is the so-called bias. One may express (380) as an equation that defines a hyperplane:

$$\boxed{\{N\}^T \{z\} + (-b) = 0}, \tag{381}$$

where $\boxed{\{w\} = \begin{Bmatrix} \{w\} \\ -1 \end{Bmatrix}}$ (382) and $\boxed{\{z\} = \begin{Bmatrix} \{x\} \\ y \end{Bmatrix}}$ (383).

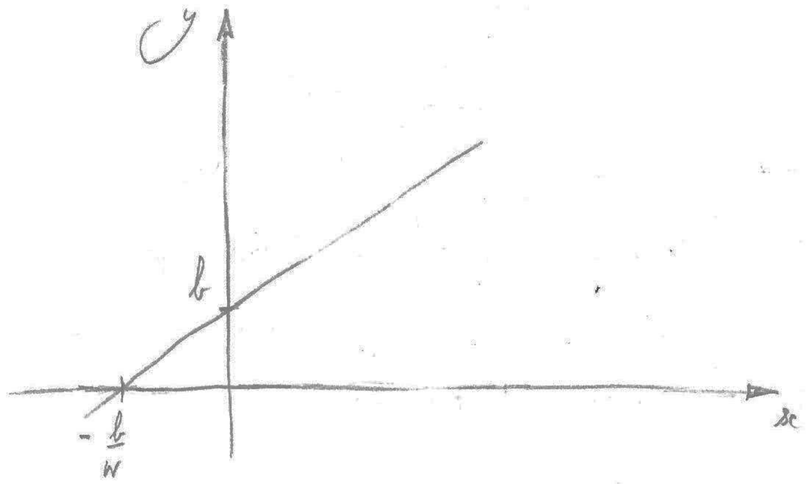
By associating one axis for which coordinate of $\{z\}$, one may show that the hyperplane defined by (381) has normal

$$(4): \quad \{m\} = \frac{\{w\}}{\|\{w\}\|} \quad (=) \quad \boxed{\{m\} = \frac{\{w\} - \{z_y\}}{\sqrt{1 + \|\{w\}\|^2}}}, \tag{384}$$

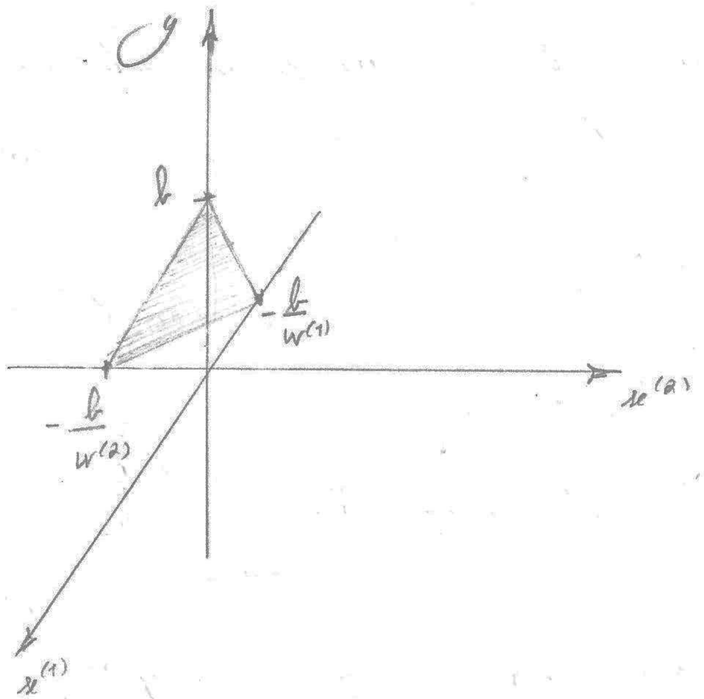
(Portion of axis associated with y)

and cross the j-th axis at $-\frac{b}{w_j} = -\frac{b}{w_j}$ if the first D axes are associated with the dimensions of $\{x\}$ and ordered in the same way (see page 4 for more details). The hyperplane also crosses the y-axis at $-\frac{b}{w_{D+1}} = -\frac{b}{-1} = b$. Examples of such hyperplanes are drawn below.

$D=1$:



$D=2$:



There is a more convenient way of expressing (380) without explicitly evidencing the bias b:

$$\boxed{y = \{\tilde{w}\}^T \{\tilde{x}\}}, \tag{385}$$

where $\boxed{\{\tilde{w}\} = \begin{Bmatrix} b \\ \{w\} \end{Bmatrix}}$ (386) and $\boxed{\{\tilde{x}\} = \begin{Bmatrix} 1 \\ \{x\} \end{Bmatrix}}$ (387). For simplicity reasons let us make the above change of notation $\{\tilde{x}\} \leftrightarrow \{x\}$ and $\{\tilde{w}\} \leftrightarrow \{w\}$, and re-write (385) as

$$\boxed{\boxed{y = \{w\}^T \{x\}}} \tag{388}$$

The Linear Regression problem is to make the hyperplane (389) to be the closest to the data points $(\{x_i\}, y_i)$. If a "least squares" sense is chosen, the respective cost function corresponds to the residual sum of squares and the Linear Regression problem is transformed into

$$\boxed{\{w^*\} = \underset{\{w\}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\{w\}^T\{x_i\} - y_i\right)^2} \tag{389}$$

Let $\boxed{[A] = \left[\{x_1\} \cdots \{x_m\}\right]}$ (390) and $\boxed{\{b\} = [y_1 \cdots y_m]^T}$ (391). One may rewrite (389) as

$$\{w^*\} = \underset{\{w\}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\underbrace{\{w\}^T[A]}_{\text{scalars}}\{e_i\} - \underbrace{\{b\}^T}_{\text{scalar}}\{e_i\}\right)^2 =$$

$$= \underset{\{w\}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\{e_i\}^T\left([A]^T\{w\} - \{b\}\right)\right)^2 =$$

$$= \underset{\{w\}}{\operatorname{argmin}} \left(\{w\}^T[A] - \{b\}^T\right)\underbrace{\left(\sum_{i=1}^{m}\{e_i\}\{e_i\}^T\right)}_{=[I]}\left([A]^T\{w\} - \{b\}\right) =$$

$$= \underset{\{w\}}{\operatorname{argmin}} \{w\}^T[A][A]^T\{w\} - \{b\}^T[A]^T\{w\} - \underbrace{\{w\}^T[A]\{b\}}_{\text{scalar}} + \{b\}^T\{b\} \Leftrightarrow$$

$$\Leftrightarrow \boxed{\{w^*\} = \underset{\{w\}}{\operatorname{argmin}} \{w\}^T[A][A]^T\{w\} - 2\{b\}^T[A]^T\{w\} + \{b\}^T\{b\}} \tag{392}$$

(Linear Regression problem)

Let $\boxed{\mathcal{L}(\{w\}) = \{w\}^T[A][A]^T\{w\} - 2\{b\}^T[A]^T\{w\} + \{b\}^T\{b\}}$ (393)

One has

$$\frac{\partial \mathcal{L}}{\partial w^{(j)}} = \underbrace{\{e_j\}^T[A][A]^T\{w\}}_{\text{scalar}} + \{w\}^T[A][A]^T\{e_j\} - 2\{b\}^T[A]^T\{e_j\}$$

$$\Leftrightarrow \boxed{\frac{\partial \mathcal{L}}{\partial w^{(j)}} = 2\left(\{w\}^T[A][A]^T - \{b\}^T[A]^T\right)\{e_j\}} \tag{394}$$

$$\left.\frac{\partial \mathcal{L}}{\partial w^{(j)}}\right|_{\{w\}=\{w^*\}} = 0 \Leftrightarrow \{w^*\}^T[A][A]^T - \{b\}^T[A]^T = \{0\}^T \Leftrightarrow$$

$$\Leftrightarrow [A][A]^T\{w^*\} - [A]\{b\} = \{0\} \Leftrightarrow \boxed{\{w^*\} = \left([A][A]^T\right)^{-1}[A]\{b\}} \tag{395}$$

(Optimum weight vector)

Note that

$$\frac{\partial^2 \mathcal{L}}{\partial w^{(i)}\partial w^{(j)}} = 2\{e_i\}^T[A][A]^T\{e_j\} \Leftrightarrow \boxed{\frac{\partial^2 \mathcal{L}}{\partial w^{(i)}\partial w^{(j)}} = 2\left([A][A]^T\right)_{ij}} \tag{396}$$

Since $[A][A]^T$ is a positive semi-definite matrix, the Hessian matrix is also positive semi-definite, and $\mathcal{L}$ is convex. Therefore, $\{w\}$ given by (395) is an absolute minimizer.

Probabilistic view of Linear Regression

One may assume that $\{x\}$ and y are related to each other so that given $\{x\}$ and $\{w\}$,

$$\boxed{y = \{w\}^T \{x\} + \varepsilon}, \tag{397}$$

where ε is some noise which is Gaussian, of mean 0 and covariance matrix $\boxed{[\Sigma] = \sigma^2 [I]}$, (398) that is,

$$(79): \boxed{f(\varepsilon) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\left(\frac{\varepsilon}{\sigma}\right)^2}}. \tag{399}$$

Note that σ is known. Well,

$$f(y = y_0 \mid \{x\}, \{w\}) \overset{(397)}{=} f(\varepsilon = y_0 - \{w\}^T \{x\} \mid \{x\}, \{w\}) \Leftarrow \left(\begin{array}{l}\text{Change of} \\ \text{notation:} \\ y_0 \leftrightarrow y\end{array}\right)$$

$$\Leftarrow f(y \mid \{x\}, \{w\}) = f(\varepsilon = y - \{w\}^T \{x\} \mid \{x\}, \{w\}) \overset{(398)}{\Leftarrow}$$

$$\Leftrightarrow \boxed{f(y \mid \{x\}, \{w\}) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\left(\frac{y - \{w\}^T \{x\}}{\sigma}\right)^2}}. \tag{400}$$

For the case of m samples $(\{x_1\}, y_1), \dots, (\{x_m\}, y_m)$, it would be convenient to re-express (400) as

$$\boxed{f(y_i \mid \{x_i\}, \{w\}) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\left(\frac{y_i - \{w\}^T \{x_i\}}{\sigma}\right)^2}}. \tag{401}$$

And from (53),

$$f(\{x_i\}, y_i \mid \{w\}) = f(y_i \mid \{x_i\}, \{w\}) \cdot f(\{x_i\}) \overset{(400)}{\Leftarrow}$$

$$\Leftarrow \boxed{f(\{x_i\}, y_i \mid \{w\}) = f(\{x_i\}) \cdot \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\left(\frac{y_i - \{w\}^T \{x_i\}}{\sigma}\right)^2}}. \tag{402}$$

The log-likelihood function associated with the experiment corresponds to

$$\log \mathcal{L}(\{w\} \mid \{x_1\}, y_1, \dots, \{x_m\}, y_m) := \log\left(f(\{x_1\}, y_1, \dots, \{x_m\}, y_m \mid \{w\})\right) =$$

$$= \log\left(\prod_{i=1}^{m} f(\{x_i\}, y_i \mid \{w\})\right) =$$

$$\left(\begin{array}{l}(\{x_1\}, y_1), \dots, (\{x_m\}, y_m) \text{ are} \\ \text{independent}\end{array}\right)$$

$$= \sum_{i=1}^{m} \log\left(f(\{x_i\}, y_i \mid \{w\})\right) \overset{(402)}{=}$$

$$= \sum_{i=1}^{m} \log\left(f(\{x_i\}) \cdot \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\left(\frac{y_i - \{w\}^T \{x_i\}}{\sigma}\right)^2}\right) \Leftarrow$$

$$\Leftarrow \boxed{\log \mathcal{L}(\{w\} \mid \mathcal{D}) = -m \log(\sqrt{2\pi}\,\sigma) + \left(\sum_{i=1}^{m} \log(f(\{x_i\}))\right) - \frac{1}{2\sigma^2}\left(\sum_{i=1}^{m} (\{w\}^T \{x_i\} - y_i)^2\right)}. \tag{403}$$

According to the Maximum Likelihood principle, one has

$$(78): \boxed{\{w^*\} = \underset{\{w\}}{\operatorname{argmax}} \log \mathcal{L}(\{w\} \mid \mathcal{D}) = \underset{\{w\}}{\operatorname{argmin}} \sum_{i=1}^{m} (\{w\}^T \{x_i\} - y_i)^2}. \tag{404}$$

Problems (404) and (389) are equivalent. Therefore, the Linear Regression problem is equivalent to find the optimum parameter $\{w\}$ according to the Maximum Likelihood principle assuming that the label y is a combination of the linear term $\{w\}^T\{x\}$ and Gaussian noise ε of null mean and isotropic (also termed spherical) covariance matrix $[\Sigma] = \sigma^2[I]$.

Ridge Regression (L2 regularisation of Linear Regression)

One should be careful when performing regressions as overfitting (the model has low bias - it predicts quite well the training data but not so much the test one) may occur, and it usually comes with a weight vector $\{w\}$ with too large components. Such large components make the model to have large variance, that is, to be too sensible to perturbations on the feature values. Indeed, if there is a perturbation $\{\varepsilon\}$ on $\{x\}$, one has from (388) that

$$y(\{x\}+\{\varepsilon\}) = \{w\}^T(\{x\}+\{\varepsilon\}) = \underbrace{\{w\}^T\{x\}}_{=y} + \underbrace{\{w\}^T\{\varepsilon\}}_{=\delta}$$

If the components of $\{w\}$ are too large, the perturbation $\delta = \{w\}^T\{\varepsilon\}$ on y caused by a perturbation $\{\varepsilon\}$ on $\{x\}$ may also be and model may significantly deviate from points not pertaining to

the training set. One way of avoiding overfitting is by considering regularisation, which reduces the complexity of the model. A simple way is through reduction of the components of $\{w\}$.

Let one consider model (397). And, also that the weight vector $\{w\}$ follows a Gaussian distribution of null mean and covariance matrix $[\Sigma] = \nu^2[I]$, that is,

$$f(\{w\}) = \frac{1}{(2\pi)^{\frac{D}{2}}\nu} e^{-\frac{1}{2}\left(\frac{\|\{w\}\|}{\nu}\right)^2} \tag{405}$$

By assuming such, one is stating that $\{w\}$ has a higher probability of being close to $\{0\}$ than far (wherein one may already infer that it is a way of imposing regularisation).

Well,

$$f(\{w\}|\mathcal{D}) \overset{\text{(Bayes' theorem, (56))}}{=} \frac{f(\mathcal{D}|\{w\})\, f(\{w\})}{f(\mathcal{D})} \overset{\text{(}(\{x_1\},y_1),\dots,(\{x_m\},y_m)\text{ are independent)}}{=} \frac{\left(\prod_{i=1}^{m} f(\{x_i\},y_i|\{w\})\right) f(\{w\})}{f(\mathcal{D})} =$$

$$\overset{(52)}{=} \frac{f(\{w\})}{f(\mathcal{D})} \prod_{i=1}^{m} f(y_i|\{x_i\},\{w\})\, f(\{x_i\}) \overset{(403),(401)}{=}$$

$$= \left| f(\{w\}|\mathcal{D}) = \frac{e^{-\frac{1}{2}\left(\frac{\|\{w\}\|}{\nu}\right)^2}}{(2\pi)^{\frac{D}{2}}\nu f(\mathcal{D})} \prod_{i=1}^{m} \frac{f(\{x_i\})}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\left(\frac{y_i - \{w\}^T\{x_i\}}{\sigma}\right)^2} \right| \tag{406}$$

From the Maximum A Posteriori principle, one gets:

$$(7.7):\ \{w^*\} = \arg\max_{\{w\}} f(\{w\}|\mathcal{D}) \Leftrightarrow \boxed{\{w^*\} = \arg\max_{\{w\}} \log\left(f(\{w\}|\mathcal{D})\right)} \quad (407)$$

$$= : f(\{w\})$$

Well,

$$f(\{w\}) = \log\left(f(\{w\}|\mathcal{D})\right) \overset{(406)}{=} \log\left(\frac{\prod_{i=1}^{m} f(\{x_i\})}{(2\pi)^{\frac{D+m}{2}} \sigma_v^m f(\mathcal{D})}\right) - \frac{1}{2}\frac{\|\{w\}\|^2}{\sigma^2}$$

$$= \mathcal{C}$$

$$-\frac{1}{2\sigma_v^2}\sum_{i=1}^{m}\left(y_i - \{w\}^T\{x_i\}\right)^2 \quad (=)$$

$$\boxed{f(\{w\}) = \mathcal{C} - \frac{1}{2\sigma_v^2}\sum_{i=1}^{m}\left(\{w\}^T\{x_i\} - y_i\right)^2 - \frac{1}{2\sigma^2}\|\{w\}\|^2} \quad (408)$$

By substituting (408) on (407) one gets:

$$\boxed{\{w^*\} = \arg\min_{\{w\}} \sum_{i=1}^{m}\left(\{w\}^T\{x_i\} - y_i\right)^2 + \left(\frac{\sigma_v}{\sigma}\right)^2\|\{w\}\|^2} \quad (409)$$

By comparing (409) with (389), one may conclude that if one further imposes $\{w\}$ to follow a Gaussian distribution of null mean and isotropic covariance matrix, the cost function is the one of the Linear Regression with an additional term associated with the norm of $\{w\}$. Therefore, not only the residual sum of squares is to be minimized but also the components of $\{w\}$.

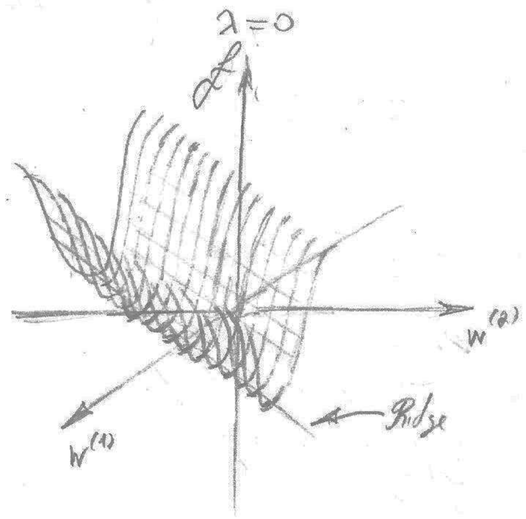
Problem (409) with $\lambda = \left(\frac{\sigma_v}{\sigma}\right)^2 \geq 0$ corresponds to an L2 regularization (since it uses the l^2-norm of $\{w\}$) of the Linear Regression problem, being termed Ridge Regression:

$$\boxed{\{w^*\} = \arg\min_{\{w\}} \sum_{i=1}^{m}\left(\{w\}^T\{x_i\} - y_i\right)^2 + \lambda\|\{w\}\|^2} \quad (410)$$

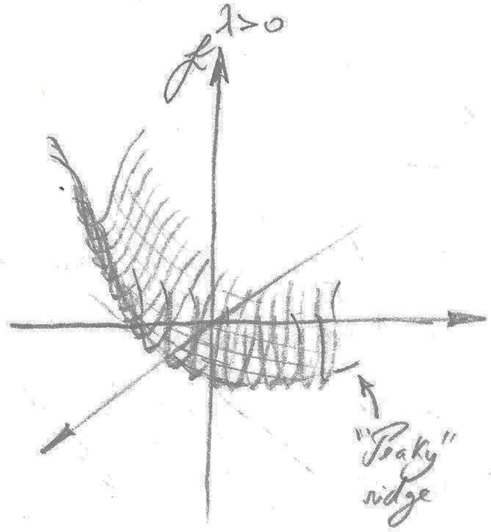
(Regularization term)

(Ridge Regression)

Let $\boxed{f(\{w\}) = \sum_{i=1}^{m}\left(\{w\}^T\{x_i\} - y_i\right)^2 + \lambda\|\{w\}\|^2}$ (411). Ridge Regression is called in that way since in the case of the cost function $f(\{w\})$ having a ridge when $\lambda = 0$ - corresponding to a line of absolute minima - is transformed into another in which the ridge is converted to a smooth peak when $\lambda > 0$ - having a sole point as an absolute minimum:



("Vanilla" Linear Regression)



(Ridge Regression)

Let one solve (410): One may understand that

$$\mathcal{L}(\{w\}) = \{w\}^T[A][A]^T\{w\} - 2\{b\}^T[A]^T\{w\} + \{b\}^T\{b\} + \lambda\{w\}^T\{w\} \tag{412}$$

(Page 212)

$$\frac{\partial \mathcal{L}}{\partial w^{(j)}} = \overbrace{\{e_j\}^T[A][A]^T\{w\}}^{\text{(scalar)}} + \{w\}^T[A][A]^T\{e_j\} - 2\{b\}^T[A]^T\{e_j\} + \overbrace{\lambda\{e_j\}^T\{w\}}^{\text{(scalar)}} + \lambda\{w\}^T\{e_j\} \quad (=$$

$$(= \boxed{\frac{\partial \mathcal{L}}{\partial w^{(j)}} = 2\left(\{w\}^T\left([A][A]^T + \lambda[I]\right) - \{b\}^T[A]^T\right)\{e_j\}} \tag{413}$$

$$\frac{\partial \mathcal{L}}{\partial w^{(j)}}\bigg|_{\{w\}=\{w^*\}} = 0 \iff \{w^*\}^T\left([A][A]^T + \lambda[I]\right) - \{b\}^T[A]^T = \{0\}^T \iff$$

$$\iff \cdot \left([A][A]^T + \lambda[I]\right)\{w^*\} = [A]\{b\} \iff \boxed{\{w^*\} = \left([A][A]^T + \lambda[I]\right)^{-1}[A]\{b\}} \tag{414}$$

(Optimum weight vector)

Lasso Regression (L1 regularisation of Linear Regression)

Instead of considering an L2 regularisation one may consider an L1 one through the so-called Lasso Regression.

"Lasso" stands for "Least Absolute Shrinkage and Selection Operator". The "Least Absolute Shrinkage" term comes from the minimisation of a cost function in which the regularisation term is the l^1-norm:

$$\{w^*\} = \underset{\{w\}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\{w\}^T\{x_i\} - y_i\right)^2 + \lambda \sum_{j=1}^{D} |w^{(j)}| \tag{415}$$

(Regularisation term)

(Lasso Regression problem)

with $\lambda \geq 0$.

The solution of (415) may not be expressed explicitly. The "Selector Operator" term comes from the fact that the minimisation of the cost function is such that it may more conveniently make a component of $\{w^*\}$ to be null, while in Ridge Regression such phenomenon may solely happen for very particular cases.

One may show that problem (415) is equivalent to

$$\{w^*\} = \underset{\{w\}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\{w\}^T\{x_i\} - y_i\right)^2$$

subject to

$$\sum_{j=1}^{D} |w^{(j)}| \leq \theta \tag{416}$$

(Lasso Regression problem)

where θ and λ related to each other.

Note: (Equivalence of (415) and (416))

From KKT theorem (Page 94), one has that (416) is equivalent to

$$\{w^*\} = \underset{\{w\}}{\operatorname{argmin}} \max_{\lambda} \sum_{i=1}^{n} \left(\{w\}^T\{x_i\} - y_i\right)^2 + \lambda\left(\sum_{j=1}^{D} |w^{(j)}| - \theta\right) \tag{417}$$

with $\lambda \geq 0$.

By changing the order of the minimisation - maximisation in (417) (which holds under the assumption of the Lagrangian being a convex function in $\{w\}$), one gets

$$\max_{\lambda} \min_{\{w\}} \sum_{i=1}^{m} (\{w\}^T \{x_i\} - y_i)^2 + \lambda \left(\sum_{j=1}^{D} |w^{(j)}| - \theta \right) \tag{418}$$

The particular problem of finding $\{w^*\}$ is equivalent to

$$\{w^*\} = \arg\min_{\{w\}} \sum_{i=1}^{m} (\{w\}^T \{x_i\} - y_i)^2 + \lambda \left(\sum_{j=1}^{D} |w^{(j)}| - \theta \right) \Leftrightarrow$$

$$\Leftrightarrow \{w^*\} = \arg\min_{\{w\}} \sum_{i=1}^{m} (\{w\}^T \{x_i\} - y_i)^2 + \lambda \sum_{j=1}^{D} |w^{(j)}| \tag{419}$$

Problem (419) corresponds to (415) which finishes the proof.

Let $f(\{w\}) = \sum_{i=1}^{m} (\{w\}^T \{x_i\} - y_i)^2$. (420) Since $\{w_{OLS}\}$ given by (395) minimizes $f(\{w\})$ (note that OLS stands for "Ordinary Least Squares" solution), one knows that

$$f(\{w\}) \geq f(\{w_{OLS}\}) \Leftrightarrow f(\{w\}) = f(\{w_{OLS}\}) + \mathcal{C} \tag{421}$$

with $\mathcal{C} \geq 0$ being some constant. It would be convenient to know the shape of the contour lines defined as

$$S_{\mathcal{C}} = \{ \{w\} \in \mathbb{R}^D : f(\{w\}) = f(\{w_{OLS}\}) + \mathcal{C} \} \tag{422}$$

From (421) and (420) one gets (Page 212)

$$\sum_{i=1}^{m} (\{w\}^T \{x_i\} - y_i)^2 = \sum_{i=1}^{m} (\{w_{OLS}\}^T \{x_i\} - y_i)^2 + \mathcal{C} \Leftrightarrow$$

$$\Leftrightarrow \{w\}^T [A][A]^T \{w\} - 2\{b\}^T [A]^T \{w\} + \cancel{\{b\}^T \{b\}} = \{w_{OLS}\}^T [A][A]^T \{w_{OLS}\} - 2\{b\}^T [A]^T \{w_{OLS}\} + \cancel{\{b\}^T \{b\}} + \mathcal{C} \Leftrightarrow$$

$$\Leftrightarrow (\{w\} - \{w_{OLS}\})^T [A][A]^T (\{w\} - \{w_{OLS}\}) + \underbrace{\{w\}^T [A][A]^T \{w_{OLS}\}}_{\text{scalar}} + \{w_{OLS}\}^T [A][A]^T \{w\}$$

$$- \{w_{OLS}\}^T [A][A]^T \{w_{OLS}\} - 2\{b\}^T [A]^T \{w\} = \{w_{OLS}\}^T [A][A]^T \{w_{OLS}\} - 2\{b\}^T [A]^T \{w_{OLS}\} + \mathcal{C} \Leftrightarrow$$

(395) (88)

$$\Leftrightarrow (\{w\} - \{w_{OLS}\})^T [A][A]^T (\{w\} - \{w_{OLS}\}) + 2\{b\}^T [A]^T \underbrace{([A][A]^T)^{-1}}_{=[I]} [A][A]^T \{w\}$$

$$- 2\{b\}^T [A]^T [A][A]^T \{w\} = 2\{b\}^T [A]^T \underbrace{([A][A]^T)^{-1}}_{=[I]} [A][A]^T \underbrace{([A][A]^T)^{-1}}_{=[I]} [A]\{b\} - 2\{b\}^T [A]^T ([A][A]^T)^{-1} [A]\{b\}$$

$$+ \mathcal{C} \Leftrightarrow$$

$$\Leftrightarrow (\{w\} - \{w_{OLS}\})^T [A][A]^T (\{w\} - \{w_{OLS}\}) + \cancel{2\{b\}^T [A]^T \{w\}} - \cancel{2\{b\}^T [A]^T \{w\}} =$$

$$= \cancel{2\{b\}^T [A]^T ([A][A]^T)^{-1} [A]\{b\}} - \cancel{2\{b\}^T [A]^T ([A][A]^T)^{-1} [A]\{b\}} + \mathcal{C} \Leftrightarrow$$

$$\Leftrightarrow \boxed{(\{w\} - \{w_{OLS}\})^T [A][A]^T (\{w\} - \{w_{OLS}\}) = \mathcal{C}} \tag{423}$$

Well, since $[A][A]^T$ is a real symmetric matrix, its diagonalizing matrix, $[Q]$, is orthogonal (see page 25) and one may express $[A][A]^T$ as

$$\boxed{[A][A]^T = [Q][\Lambda][Q]^T} \quad (425)$$

And (423) becomes

$$(\{w\} - \{w_{OLS}\})^T [Q][\Lambda][Q]^T (\{w\} - \{w_{OLS}\}) = C \Leftrightarrow$$

$$\Leftrightarrow \underbrace{([Q]^T\{w\}}_{=:\{u\}} - \underbrace{[Q]^T\{w_{OLS}\}}_{=:\{\mu_{OLS}\}})^T [\Lambda] ([Q]^T\{w\} - [Q]^T\{w_{OLS}\}) = C \Leftrightarrow$$

$$\Leftrightarrow \boxed{(\{u\} - \{\mu_{OLS}\})^T \frac{1}{C} [\Lambda] (\{u\} - \{\mu_{OLS}\}) = 1} \quad (426)$$

where $\boxed{\{u\} = [Q]^T\{w\}}$ (427) and $\boxed{\{\mu_{OLS}\} = [Q]^T\{w_{OLS}\}}$ (428)

Equation (426) represents a hyperellipse centred at $\{u\} = \{\mu_{OLS}\}$ and semi-axes of lengths $\sqrt{\frac{C}{\lambda_j}}$, $j = 1, \dots, D$. Indeed,

$$(426): (\{u\} - \{\mu_{OLS}\})^T \frac{1}{C} [\Lambda] (\{u\} - \{\mu_{OLS}\}) = 1 \Leftrightarrow$$

$$\Leftrightarrow \sum_{j=1}^{D} \sum_{l=1}^{D} \frac{\lambda_j \delta_{jl}}{C} (u^{(j)} - \mu_{OLS}^{(j)})(u^{(l)} - \mu_{OLS}^{(l)}) = 1 \Leftrightarrow$$

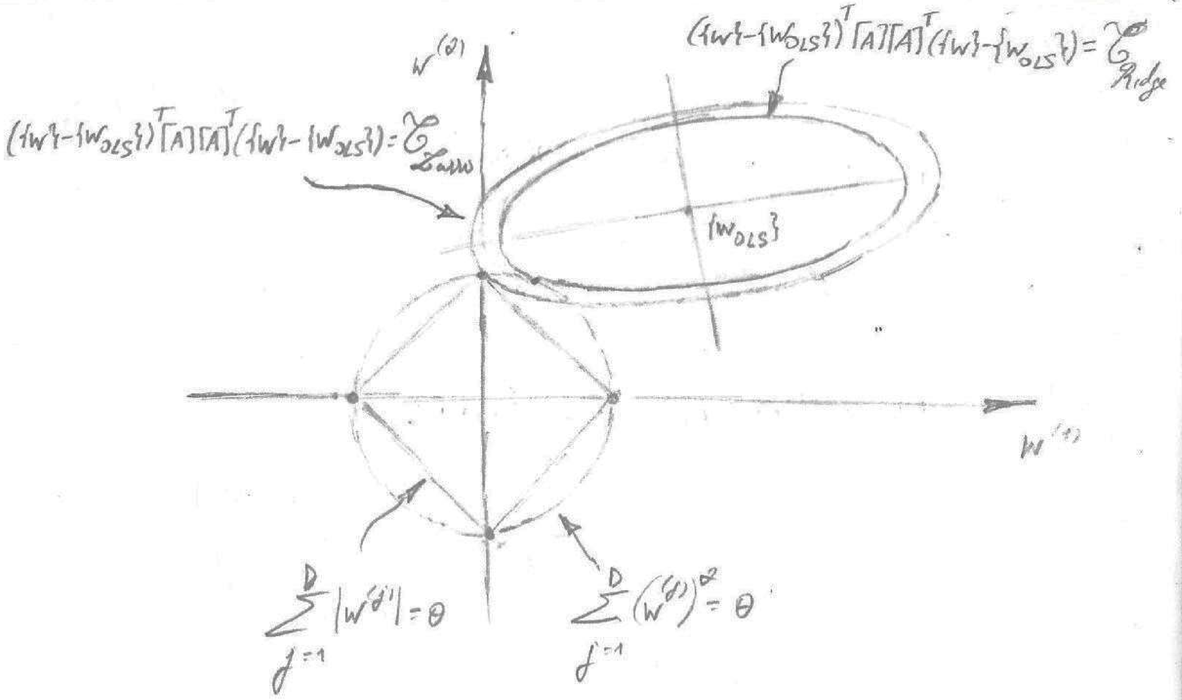
$$\Leftrightarrow \boxed{\sum_{j=1}^{D} \left(\frac{u^{(j)} - \mu_{OLS}^{(j)}}{\sqrt{\frac{C}{\lambda_j}}} \right)^2 = 1} \quad (429)$$

And since $[Q]$ is an orthogonal matrix, its application into a vector gives a rotation or reflection. Indeed, the inner product between two vectors is conserved under such transformation:

$$([Q]^T\{u\})^T ([Q]^T\{v\}) = \{u\}^T \underbrace{[Q][Q]^T}_{=[I]} \{v\} = \{u\}^T \{v\}$$

$$= [Q]^{-1}$$

Therefore, one may say that (426) represents a hyperellipse centred at $\{w\} = \{w_{OLS}\}$ and semi-axes of length $\sqrt{\frac{C}{\lambda_j}}$, $j = 1, \dots, D$, subject to some rotation or reflection around the centre. When C increases so does the semi-axes lengths. If $\{w_{OLS}\} \neq \{0\}$ then, one may conclude that an increase of C would make the hyperellipses to get closer and closer to $\{0\}$. The optimum value $\{w^*\}$ of (426) would be $\{w\}$ that pertains to the lozenge defined by the constraint $\sum_{j=1}^{D} |w^{(j)}| \leq \Theta$, being the closest to the centre of the hyperellipse. In the case of Ridge Regression one could show that the lozenge would in turn give place to a circle. For both cases, if $\{w_{OLS}\} \neq \{0\}$, the $\{w^*\}$ point would correspond to the intersection of the constraint boundary with the hyperellipse boundary, as shown in the picture below.



From the above picture, one may understand that the hyperellipse is more likely to "hit" a corner of the lozenge of the Lasso constraint than the one of the points of the circumference of the Ridge constraint at the axes of $\{w\}$ (indeed, the latter would only happen if the hyperellipse centre were at one of the axis and one of the semi-axis had the direction of such axis). That means that the optimum weight vector in the case of the Lasso Regression is more likely to have a null component, that is, $w^{*(j)}=0$, than in the case of the Ridge Regression, which further simplifies the resultant model.

Kernel Regression

One may show that Linear Regression may be kernelized. First of all, let one show that the optimum weight vector as given by (395) may be assumed to correspond to a a linear combination of the features $\{x_i\}$:

$$\{w^*\}=\sum_{i=1}^{m}\alpha_i^*\{x_i\}=\sum_{i=1}^{m}\alpha_i^*[A]\{e_i\}=[A]\underbrace{\sum_{i=1}^{m}\alpha_i^*\{e_i\}}_{=:\{\alpha^*\}} \Rightarrow$$

$$\Rightarrow \boxed{\{w^*\}=[A]\{\alpha^*\}} \tag{430}$$

where $\{\alpha^*\}$ is the optimum vector of coefficients of the linear combination. One may say that in general the weight vector $\{w\}$ is a sum of a linear combination of $\{x_i\}$ - let this be $\{w_{\parallel}\}=[A]\{\alpha\}$ - and a vector that lies out of the space spanned by $\{x_i\}$ - let this be $\{w_\perp\}$:

$$\boxed{\{w\}=\{w_{\parallel}\}+\{w_\perp\}} \tag{431}$$

By inserting (431) into the cost function of (389) one gets

$$\sum_{i=1}^{m}(\{w\}^T\{x_i\}-y_i)^2=\sum_{i=1}^{m}(\{w_{\parallel}\}^T\{x_i\}+\underbrace{\{w_\perp\}^T\{x_i\}}_{\substack{=0 \text{ since } \{w_\perp\} \\ \text{is orthogonal with} \\ \text{respect to any } \{x_i\}}}-y_i)^2=$$

$$=\sum_{i=1}^{m}(\{w_{\parallel}\}^T\{x_i\}-y_i)^2$$

The presence of $\{w_1\}$ in the cost function is irrelevant to its value and one may assume that $\{w_1\} = \{0\}$ without loss of generality. Let one further level of (430) with the help of (395):

$$[A]\{\alpha^*\} = ([A][A]^T)^{-1}[A]\{b\} \Leftrightarrow$$

$$\Leftrightarrow [A]^T[A][A]^T[A]\{\alpha^*\} = \underbrace{[A]^T[A][A]^T([A][A]^T)^{-1}}_{=[I]}[A]\{b\} \Leftrightarrow$$

$$\Leftrightarrow ([A]^T[A])^2\{\alpha^*\} = [A]^T[A]\{b\} \Leftrightarrow \boxed{\{\alpha^*\} = ([A]^T[A])^{-1}\{b\}} \quad (432)$$

It could then be convenient to use mappings $\{\varphi\}$ associated with kernel functions k given by (339) and then to use the Gram matrix

$$\boxed{[K] = [A]^T[A]}, \quad (433)$$

where $\boxed{[A] = \left[\{\varphi(\{x_1\})\} \cdots \{\varphi(\{x_m\})\}\right]}$. (434) Therefore, (432) becomes

$$\boxed{\{\alpha^*\} = [K]^{-1}\{b\}} \quad (435)$$

The Linear Regression model (388) is transformed into

$$y = \{w^*\}^T\{\varphi(\{x\})\} \overset{(430)}{=} \{\alpha^*\}^T[A]^T\{\varphi(\{x\})\} \overset{(435)}{=} \{b\}^T[K]^{-1}[A]^T\{\varphi(\{x\})\} \overset{(434)}{\Leftrightarrow}$$

$$\Leftrightarrow y = \{b\}^T[K]^{-1}\begin{Bmatrix} \{\varphi(\{x_1\})\}^T\{\varphi(\{x\})\} \\ \vdots \\ \{\varphi(\{x_m\})\}^T\{\varphi(\{x\})\} \end{Bmatrix} = \{b\}^T\sum_{j=1}^{m}\sum_{l=1}^{m}([K]^{-1})_{jl}\{\varphi(\{x$$

$$= \{b\}^T\sum_{i=1}^{m}\sum_{j=1}^{m}([K]^{-1})_{ij}\underbrace{\{\varphi(\{x_j\})\}^T\{\varphi(\{x\})\}}_{=k(\{x_j\},\{x\})}\{l_i\} \overset{(391)}{\Leftarrow}$$

$$\Leftrightarrow \boxed{y = \sum_{i=1}^{m} y_i\left(\sum_{j=1}^{m}([K]^{-1})_{ij}\, k(\{x_j\},\{x\})\right)} \quad (436)$$

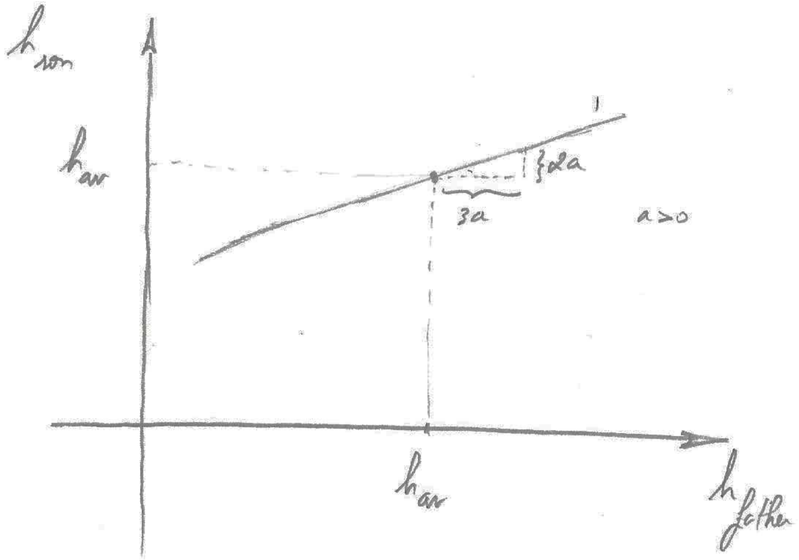
(Kernel Regression)

Note that a Kernel Regression is non-parametric. The model solely depends on the parameters of the kernel function $k(\{x\},\{z\})$ – which are chosen by the analyst, and are, therefore, hyperparameters.

Note: (Etymology of "regression")

The term "regression" came from the historic usage of Linear Regression by Francis Galton in 1886 to model "regression toward the mean" for the case of a father of height greater than the average one having a son that is taller than the average by 2/3 of the difference between its father's height and that average. The analogous would occur for the case of a shorter than average father.

Such result may be interpreted from a plot of fathers' heights in the x axis and sons' heights in the y axis. A line of slope $w = \frac{2}{3}$ that crosses the point (h_{av}, h_{av}), where h_{av} is the mean average height, would be obtained:



VIII - K - Nearest Neighbours

K-Nearest Neighbours (KNN) is a classification algorithm that defines the label y of a given feature $\{x\}$ as the mode of the labels y_i of the k features $\{x_i\}$ from the dataset D that are more similar (according to some metric) to $\{x\}$. Here in "mode" is the term that appears more times in the set of labels of the k most similar features.

Let $(\{x_1^*\}, y_1^*), \ldots (\{x_k^*\}, y_k^*)$ be the k points whose features are the most similar to $\{x\}$. The KNN algorithm is such that

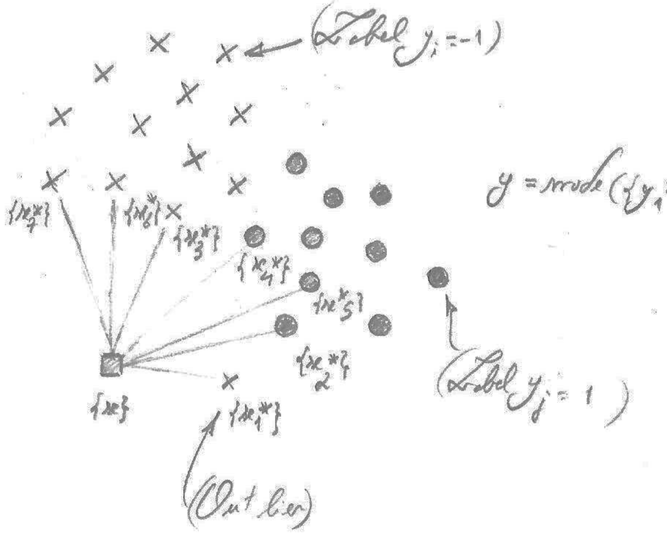
$$\boxed{y = \text{mode}\,(\{y_1^*, \ldots, y_k^*\})} \tag{437}$$

Let s be the similarity metric. One common choice is the negative of the l^2-norm (negative of the Euclidian distance):

$$\boxed{s(\{x_i\}, \{x\}) = -\|\{x_i\} - \{x\}\| = -\sqrt{\sum_{j=1}^{D} \left(x_i^{(j)} - x^{(j)}\right)^2}} \tag{438}$$

Similar $\{x_i\}$ and $\{x\}$ features have a higher $s(\{x_i\}, \{x\})$ value.

An example of the application using the negative of the L^2-norm metric with $K=7$, is depicted below.

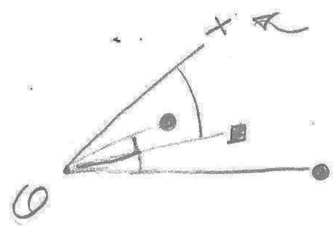


From such example, one may understand why it is important to pick the mode of the labels of the K most similar features instead of solely the label of the most similar one: if the most similar one is an outlier (see $\{x_6^*\}$) its label might not be the most appropriate one for the feature $\{x\}$. Also if more points are to be labelled considering $k=1$ and a new point was already labelled using an outlier, clusters of outliers may start to appear when using the KNN algorithm, which is utterly undesirable.

There are other similarity metrics such as the cosine similarity which is as greater as lesser is the angle between the vectors:

$$s(\{x_i\},\{x\}) = \cos\left(\angle(\{x_i\},\{x\})\right) = \left(\frac{\{x_i\}}{\|\{x_i\}\|}\right)^T \frac{\{x\}}{\|\{x\}\|} \rightarrow$$

$$\rightarrow \cos\left[s(\{x_i\},\{x\}) = \frac{\sum_{j=1}^{D} x_i^{(j)} x^{(j)}}{\sqrt{\sum_{j=1}^{D}\left(x_i^{(j)}\right)^2} \cdot \sqrt{\sum_{j=1}^{D}\left(x^{(j)}\right)^2}} \right] \tag{439}$$



Note that both the number K and the metric's needs to be defined by the analyst and are, therefore, hyperparameters. Also, note that the KNN algorithm is of the type "instance-based" in contrast with "model-based" since it does not construct any model — there is nothing to train — it solely labels new points, one by one, that is, "instance by instance," using the whole previous data. For the case of a model, the original data may be disregarded but not in the case of an algorithm such as KNN.

IX - Decision Trees

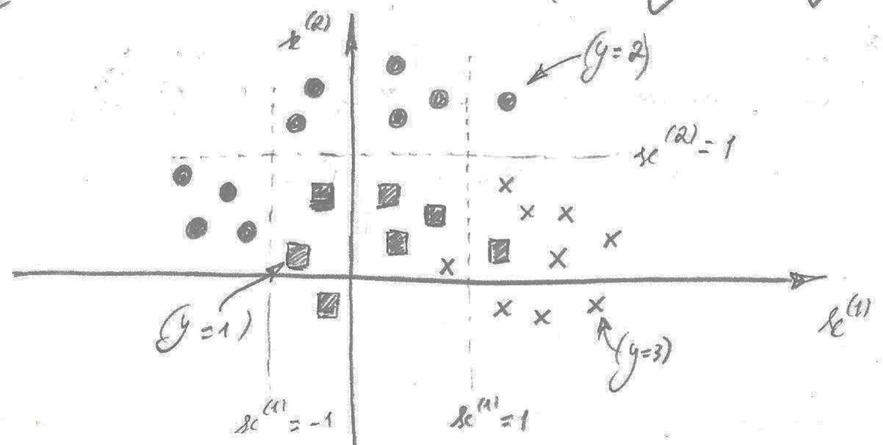
A decision tree is typically used for classification. It predicts the label y of a feature x by determining which subset of training data points it pertains to. The predicted label would be the mode of those associated with the features of the subset. Each subset is obtained by imposing discriminative rules to the feature values so that at the end the subset is close to "pure" - in the sense that the labels of the features pertaining to the subset are almost all the same. By making the subset to be almost "pure", in the case of the test feature "falling" into it, there would be a greater confidence on its label being the same as the mode of the training data pertaining to it.

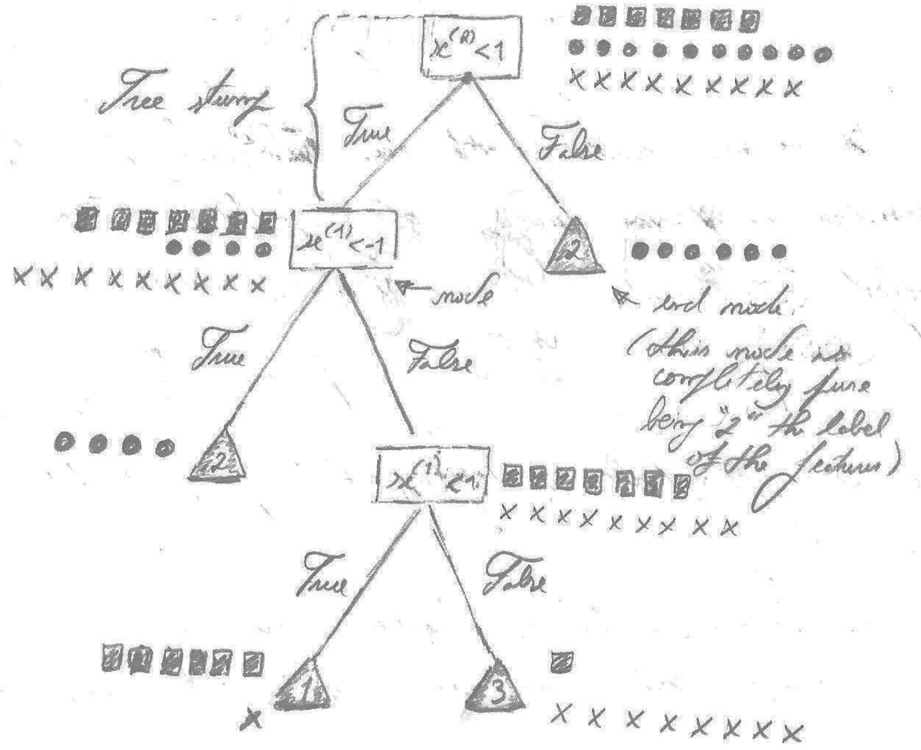
An example of an algorithm for constructing the decision tree is ID3 - "Iterative Dichotomiser 3" (the number 3 is simply the version of the algorithm). The term "dichotomiser" comes from the fact that discriminations on the feature values, which are inequalities of the form $x^{(j)} < t$ where t is some threshold results in two possible paths (a dichotomy) - one for which the condition holds true and another for which it does not. These discriminations are made at the so-called "nodes".

The term "iterative" comes from the fact that the discriminations are made iteratively – from one to another – making the tree to grow in nodes and branches. Each new discriminative condition would correspond to the ones that result in "purer" subsets. After a certain stopping criteria are met the so-called "end nodes" (or "leaf nodes") are formed. There associate the mode of the labels of training data of the respective subset to any testing feature that "falls" into it. Indeed, the testing feature traverses the decision tree going through the branch whose discriminative conditions are satisfied.

Let one consider the example of $\{x\} \in \mathbb{R}^2$ and $y \in \{1,2,3\}$. Let $y = 1$, 2 and 3 be associated with squares (▪), circles (●) and crosses (×), respectively.

An ID3 decision tree for the dataset presented below would look like the one in the figure following it





The ID3 algorithm chooses the index of the component j in the discriminative condition

$$\boxed{x^{(j)} < t} \qquad (140)$$

(Decision tree's discriminative condition)

and the the threshold t as the ones that maximize the "purerness" of the child subsets. The mathematical formulation of such purerness requires a former into Section on the so-called Shannon information of an event.

Shannon information

Let X be some random variable and $P_X(x)$ the probability of the outcome $X=x$ occurring. Let $I_X(x)$ be the information that the outcome x yields. This is also called "information content", "surprisal", "Shannon information" or "self-information" that contrasts with the mutual information $I(X;Y)$ - to be presented later.

According to Claude Shannon, the information associated with an event is a solo function of its probability and

1- An event with probability $P_X(x)=1$ is perfectly unsurprising, and therefore, it yields no information, that is,

$$\boxed{I_X(x)=0, \text{ if } P_X(x)=1} \tag{441}$$

2- The less probable an event is, the more surprising it is and the more information it yields, that is,

$$\boxed{I_X(x) > I_Y(y) \text{ if } P_X(x) < P_Y(y)} \tag{442}$$

3- If two independent events are measured separately, the total amount of information is the sum of the self-information of the individual events, that is,

$$\boxed{I_{X,Y}(x,y) = I_X(x) + I_Y(y) \text{ if } X \perp Y} \tag{443}$$

One function that satisfies (441), (442) and (443) is

$$\boxed{I_X(x) = -k \log\left(P_X(x)\right)} \tag{444}$$

where $k>0$ is a constant. This constant would need to be arbitrated. It is possible to show that (444) is equivalent to

$$\boxed{I_X(x) = -\log_b\left(P(x)\right)} \tag{445}$$

(Shannon information)

where $b>1$ is some base. Indeed,

$$\log_b\left(P_X(x)\right) = \frac{\log_a\left(P_X(x)\right)}{\log_a b} = \underbrace{\frac{1}{\log_a b}}_{=:k} \log\left(P_X(x)\right) = k \log\left(P_X(x)\right)$$

[Note: (Change of base of logarithm)

Let one show that

$$\boxed{\log_b x = \frac{\log_a x}{\log_a b}} \tag{446}$$

$$\log_a x = \log_a\left(b^{\log_b x}\right) = \left(\log_b x\right)\left(\log_a b\right) \quad (\text{as } x = b^{\log_b x})$$

$$\Rightarrow \log_b x = \frac{\log_a x}{\log_a b} \quad \square$$]

Let one arbitrate $b = e$.

Shannon entropy

The Shannon entropy or "information entropy" is the expected value of the information associated with a variable:

$$\mathcal{H}(X) := \mathbb{E}(I_x) = \sum_{x \in \mathcal{S}_x} P_x(x)\, I_x(x) \quad (= \overset{(445)}{\leftarrow}$$

$$(= \boxed{\mathcal{H}(X) = -\sum_{x \in \mathcal{S}_x} P_x(x) \log\left(P_x(x)\right)} \quad (447)$$

(Shannon entropy)

where $\mathcal{S}_x$ is the set of values that X may take. Note this entropy was previously introduced, corresponding to (158).

For the case of $\mathcal{S}_x$ corresponding to two elements (for instance the case of a coin $f(t)$), let this be $\mathcal{S} = \{0,1\}$, with $P(X=1) = f$, the entropy would become

$$(447) \quad \boxed{\mathcal{H}(X) = -\left(f \log(f) + (1-f)\log(1-f)\right)} \quad (448)$$

(Shannon entropy for the case of X only possibly assuming two values, where f is the probability of X being some

For such case $\mathcal{H}(X)$ is null for $f=0$ and $f=1$ (since there would correspond to constancy of $X=0$ and $X=1$, respectively). Indeed,

$$\mathcal{H}(X)\Big|_{f=0} = -\left(0\log(0) + 1\underbrace{\log(1)}_{=0}\right) = -\lim_{f\to 0} f\log f =$$

$$= -\lim_{f\to 0} f\left((f-1) + O(f-1)^2\right) = -\lim_{f\to 0} f(f-1) + f\,O(f-1)^2 =$$

(Taylor series expansion of $\log f$ around $f=1$)

$$= 0.$$

$$\mathcal{H}(X)\Big|_{f=1} = -\left(1\underbrace{\log(1)}_{=0} + 0\log(0)\right) = 0$$

Also, $\mathcal{H}(X)$ is a symmetric function with respect to $f=\frac{1}{2}$:

$$\mathcal{H}(X)\Big|_{f=\frac{1}{2}+q} = -\left(\left(\tfrac{1}{2}+q\right)\log\left(\tfrac{1}{2}+q\right) + \left(\tfrac{1}{2}-q\right)\log\left(\tfrac{1}{2}-q\right)\right)$$

$$\mathcal{H}(X)\Big|_{f=\frac{1}{2}-q} = -\left(\left(\tfrac{1}{2}-q\right)\log\left(\tfrac{1}{2}-q\right) + \left(\tfrac{1}{2}+q\right)\log\left(\tfrac{1}{2}+q\right)\right) =$$

$$= \mathcal{H}(X)\Big|_{f=\frac{1}{2}+q}$$

And the entropy function is concave:

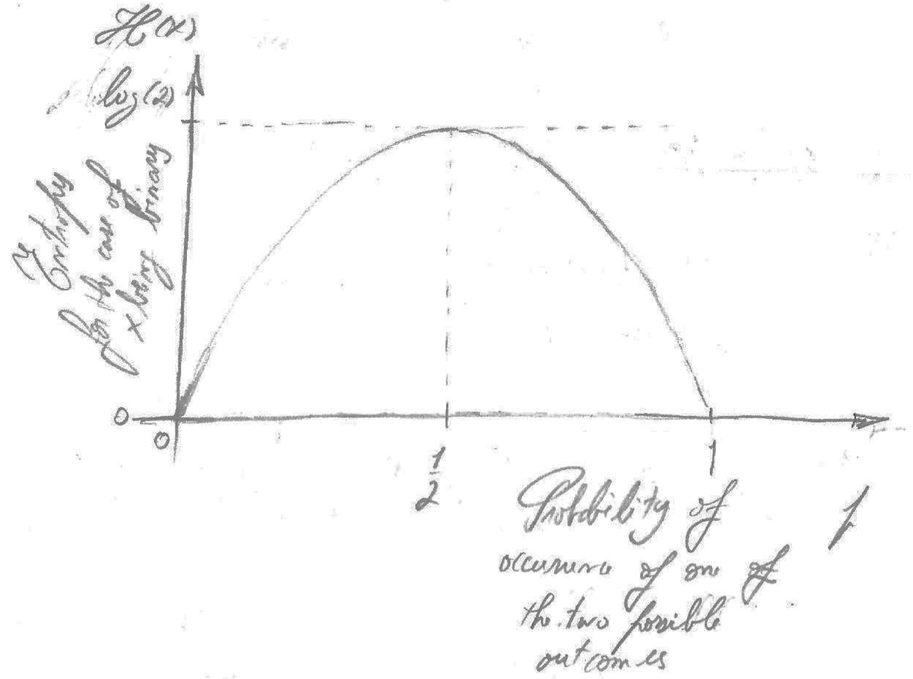
$$\frac{\partial \mathcal{H}}{\partial f} = -\left(\log(f) + 1 - \log(1-f) - 1\right) =$$

$$= -\log\frac{f}{1-f}$$

$$\frac{\partial^2 \mathcal{H}}{\partial f^2} = -\frac{\frac{1-f+f}{(1-f)^2}}{\frac{f}{1-f}} = -\frac{1}{f(1-f)}$$

>0, since $f \in [0,1]$.

Therefore $\mathcal{H}(X)$ as a maximum at $f = \frac{1}{2}$. The plot $\mathcal{H}(X)$ versus f would be the one presented below.



Shannon conditional entropy

Shannon conditional entropy of X given Y is the expected value in (X,Y) of the information of event X given event Y, that is,

$$\mathcal{H}(X|Y) = \mathbb{E}_{(X,Y)\sim P_{X,Y}}(I_{X|Y}) = \sum_{x\in\mathcal{S}_X}\sum_{y\in\mathcal{S}_Y} P_{X,Y}(x,y)\, I_{X|Y} =$$

$$\overset{(445)}{=} -\sum_{x\in\mathcal{S}_X}\sum_{y\in\mathcal{S}_Y} P_{X,Y}(x,y)\log\left(P_{X|Y}(x,y)\right) \overset{(52)}{\Longrightarrow}$$

$$\Longrightarrow \boxed{\mathcal{H}(X|Y) = -\sum_{x\in\mathcal{S}_X}\sum_{y\in\mathcal{S}_Y} P_{X,Y}(x,y)\log\left(\frac{P_{X,Y}(x,y)}{P_Y(y)}\right)} \quad (449)$$

(Shannon conditional entropy)

Mutual information (or "information gain")

The mutual information of two random variables X and Y is the difference between the expected value of the information given by one of the variables and the expected value of the information given by that variable when given the other, that is,

$$\boxed{I(X;Y) = \mathcal{H}(X) - \mathcal{H}(X|Y) = \mathcal{H}(Y) - \mathcal{H}(Y|X)} \quad (450)$$

Mutual information is commonly referred to as information gain, since it corresponds to the expectation on the information gain when going from $X|Y$ to (X,Y).

One may further show that the mutual information $I(X;Y)$ is equivalent to the relative entropy from the product of the probabilities of X and Y, that is, $P_X \cdot P_Y$, to the joint probability of X and Y, that is, $P_{X,Y}$:

$$\boxed{I(X;Y) = \mathcal{D}(P_{X,Y} \,\|\, P_X \cdot P_Y)} \quad (451)$$

(Mutual information of X and Y)

where $\mathcal{D}$ is the relative entropy function given by (139). From (451) one understands that the mutual information corresponds to the relative entropy from $P_{X,Y}$ as if $X \perp Y$ to the actual $P_{X,Y}$. The dependence between X and Y is the factor behind the gain of information.

Let one show (451):

$$I(X;Y) = \mathcal{H}(X) - \mathcal{H}(X|Y) \overset{(447),(449)}{=}$$

$$= -\sum_{x \in \Phi_X} P_X(x) \log\left(P_X(x)\right) + \sum_{x \in \Phi_X} \sum_{y \in \Phi_Y} P_{X,Y}(x,y) \log\left(\frac{P_{X,Y}(x,y)}{P_Y(y)}\right) \quad (*)$$

$$\left(\overset{(49)}{=} \sum_{y \in \Phi_Y} P_{X,Y}(x,y)\right)$$

$$\boxed{I(X;Y) = \sum_{x \in \Phi_X} \sum_{y \in \Phi_Y} P_{X,Y}(x,y) \log\left(\frac{P_{X,Y}(x,y)}{P_X(x) \cdot P_Y(y)}\right)} \quad (452)$$

Result (452) is indeed the same as (451). From (451) one may infer that the mutual information function is symmetric, that is,

$$\boxed{I(X;Y) = I(Y,X)} \quad (453)$$

(Symmetry of the mutual information function)

And from (142), one concludes that the mutual information is null if $X \perp Y$, that is,

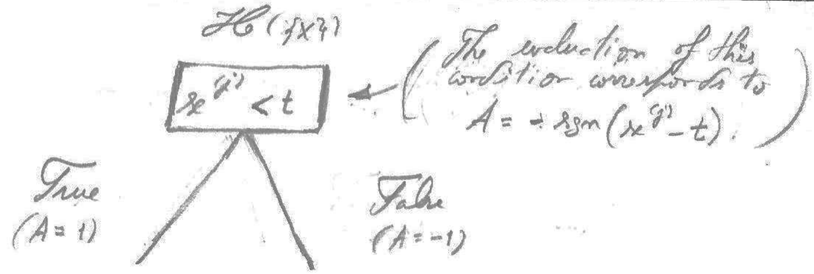
$$\boxed{I(X;Y) = 0 \quad \text{if } X \perp Y} \quad (454)$$

The ID3 algorithm

The ID3 algorithm consists in

1- Define a set of threshold values, t, for each component of the feature vector (for example evenly spaced values over the range associated with the component of the feature).

2- Pick the inequality condition that has the largest information gain:



(The evaluation of this condition corresponds to $A = +\operatorname{sgn}(x^{(j)} - t)$.)

$$I(\{x^{(j)}\}, A) = \mathcal{H}(\{x^{(j)}\}) - \mathcal{H}(\{x^{(j)}\}|A) =$$

$$= -\sum_{\{x^{(j)}\} \in \Phi_{\{x^{(j)}\}}} P_{\{x^{(j)}\}}(\{x^{(j)}\}) \log\left(P_{\{x^{(j)}\}}(\{x^{(j)}\})\right) -$$

$$+ \sum_{\{x^{(j)}\} \in \Phi_{\{x^{(j)}\}}} \sum_{a \in \Phi_A} P_{\{x^{(j)}\}, A}(\{x^{(j)}\}, a) \log\left(P_{\{x^{(j)}\}|A}(\{x^{(j)}\}, a)\right) (=)$$

Let $\mathcal{D}$ be the set of data points before the bifurcation, and let $\mathcal{D}_-$ and $\mathcal{D}_+$ be the sets of data points of $\mathcal{D}$ that satisfy $A=-1$ and $A=+1$, respectively. Furthermore, let $\mathcal{L}(\{x^{(j)}\})$ be the occurrence frequency of $\{x^{(j)}\} = \{x^{(j)}\}$ in $\mathcal{D}$. One has,

$$\left| P_{\{x^{(j)}\}}(\{x^{(j)}\}) \approx \frac{\mathcal{L}(\{x^{(j)}\})}{|\mathcal{D}|} \right| \quad (455)$$

$$\left| P_{\{x^{(j)}\}|A}(\{x^{(j)}\}, -1) \approx \frac{\mathcal{L}(\{x^{(j)}\})}{|\mathcal{D}_-|} \delta_{A(\{x^{(j)}\}), -1} \right| \quad (456)$$

$$\text{and} \left| P_{\{x^{(j)}\}|A}(\{x^{(j)}\}, +1) \approx \frac{\mathcal{L}(\{x^{(j)}\})}{|\mathcal{D}_+|} \delta_{A(\{x^{(j)}\}), +1} \right| \quad (457)$$

Also,

$$\left| P_{\{x^{(j)}\}, A}(\{x^{(j)}\}, -1) \approx \frac{\mathcal{L}(\{x^{(j)}\})}{|\mathcal{D}|} \delta_{A(\{x^{(j)}\}), -1} \right|, \quad (458)$$

$$\left| P_{\{x^{(j)}\}, A}(\{x^{(j)}\}, +1) \approx \frac{\mathcal{L}(\{x^{(j)}\})}{|\mathcal{D}|} \delta_{A(\{x^{(j)}\}), +1} \right| \quad (459)$$

$$(=) \quad I(\{x^{(j)}\}, A) = -\sum_{\{x^{(j)}\} \in \Phi_{\{x^{(j)}\}}} \frac{\mathcal{L}(\{x^{(j)}\})}{|\mathcal{D}|} \log\left(\frac{\mathcal{L}(\{x^{(j)}\})}{|\mathcal{D}|}\right)$$

$$+ \sum_{\{x^{(j)}\} \in \Phi_{\{x^{(j)}\}}} \frac{\mathcal{L}(\{x^{(j)}\})}{|\mathcal{D}|} \delta_{A(\{x^{(j)}\}), -1} \log\left(\frac{\mathcal{L}(\{x^{(j)}\})}{|\mathcal{D}_-|} \delta_{A(\{x^{(j)}\}), -1}\right)$$

$$+ \sum_{\{x^{(j)}\} \in \Phi_{\{x^{(j)}\}}} \frac{\mathcal{L}(\{x^{(j)}\})}{|\mathcal{D}|} \delta_{A(\{x^{(j)}\}), +1} \log\left(\frac{\mathcal{L}(\{x^{(j)}\})}{|\mathcal{D}_+|} \delta_{A(\{x^{(j)}\}), +1}\right) (=)$$

$$(=) \left| I(\{x^{(j)}\}, A) = \left(\sum_{\{x^{(j)}\} \in \Phi_{\{x^{(j)}\}}} \frac{\mathcal{L}(\{x^{(j)}\})}{|\mathcal{D}|} \Big/ \delta_{A(\{x^{(j)}\}), -1} \log\left(\frac{\mathcal{L}(\{x^{(j)}\})}{|\mathcal{D}_-|} \delta_{A(\{x^{(j)}\}), -1}\right) \right.\right.$$

$$\left. + \delta_{A(\{x^{(j)}\}), +1} \log\left(\frac{\mathcal{L}(\{x^{(j)}\})}{|\mathcal{D}_+|} \delta_{A(\{x^{(j)}\}), +1}\right) \right.$$

$$\left.\left. - \log\left(\frac{\mathcal{L}(\{x^{(j)}\})}{|\mathcal{D}|}\right) \right) \right| \quad (455)$$

(Information gain from the children nodes to the parent one in an ID3 bifurcation)

One may also express (455) as

$$\left| I(\{x^{(j)}\}, A) = \mathcal{H}(\mathcal{D}) - \left(\frac{|\mathcal{D}_-|}{|\mathcal{D}|} \mathcal{H}(\mathcal{D}_-) + \frac{|\mathcal{D}_+|}{|\mathcal{D}|} \mathcal{H}(\mathcal{D}_+) \right) \right| \quad (456)$$

4- Repeat the procedure for D_- and D_+

The iterative process into a branch of the tree may be stopped when some criteria are met:

- The information gain of a following step is lower than some tolerance, that is,

$$I(f_i X_j; A) < \varepsilon$$

- The depth of the branch is bigger than some value
- All feature indices j and thresholds t have been tried.

X - Logistic Regression

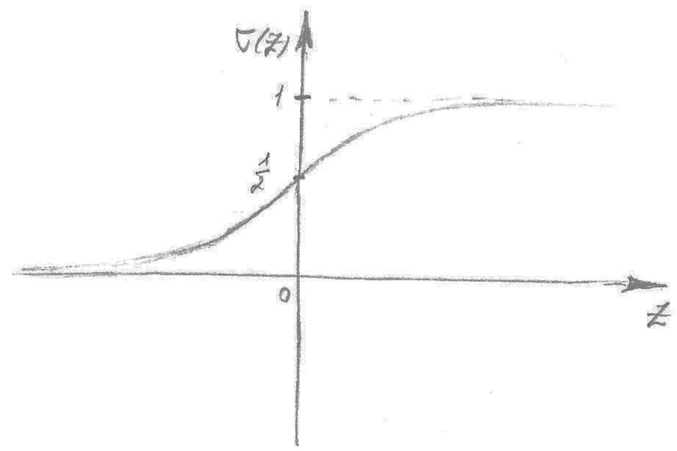
Binary Logistic Regression

Let one consider the case of binary classification in which $y \in \{-1, 1\}$. In Logistic Regression, the probability of feature $\{x\}$ being associated with the label $y=1$, is modelled through the so-called "sigmoid" function - also termed "logistic" function.

$$P(y=1) = \sigma(\{w\}^T \{x\} + b) = \frac{1}{1 + e^{-(\{w\}^T \{x\} + b)}} \quad (457)$$

(Sigmoid function of argument $z = \{w\}^T \{x\} + b$)

where $\{w\}$ is a weight vector and b a bias. The sigmoid function $\sigma(z)$ has the shape showed in the figure below.



$$\frac{d^2\sigma}{dz^2} = \frac{-e^{-z}(1+e^{-z})^2 - e^{-z}\cdot 2\cdot(-1)e^{-z}(1+e^{-z})}{(1+e^{-z})^4} \Leftarrow$$

$$= \frac{e^{-z}}{(1+e^{-z})^3}\left(2e^{-z} - (1+e^{-z})\right) \Leftarrow$$

$$\frac{d^2\sigma}{dz^2} = \frac{e^{-z}}{(1+e^{-z})^3}\left(e^{-z}-1\right)$$

$$\frac{d^2\sigma}{dz^2} = \begin{cases} >0 & \text{if } z<0 \\ =0 & \text{if } z=0 \\ <0 & \text{if } z>0 \end{cases} \Rightarrow$$ The sigmoid function is convex for $z<0$ and concave for $z>0$

Prediction of the label y for the feature x_i is such that $y=-1$ is given if $P(y=1)<\frac{1}{2}$ and $y=1$ is given if $P(y=1)>\frac{1}{2}$, that is

$$\boxed{y(x_i) = \operatorname{sgn}\left(\sigma(w^T x_i + b) - \frac{1}{2}\right)} \quad (458)$$

(Prediction of the Binary Logistic Regression)

The sigmoid function is called in that way since it has an "S" shape. And the alternative term "logistic" was coined by Pierre François Verhulst (who was the first one deriving it). He didn't explain its etymology, but it is presumed that it was due to a contrast between the curve and the exponential one - as presented by him - which at the time (~1830) was called logarithmic curve. The term "logistic" should then come from "logarithm". The term "regression" is ambiguous since the method is used for classification and not regression. But it should be due to the fact that the logistic function embeds the term $w^T x_i + b$ in it which is the model that is used for the label y in the well-known Linear Regression problem.

Let one further describe the logistic function $\sigma(z)$.

$$\sigma(z) = \frac{1}{1+e^{-z}} > 0$$

$$\lim_{z\to-\infty} \sigma = 0$$

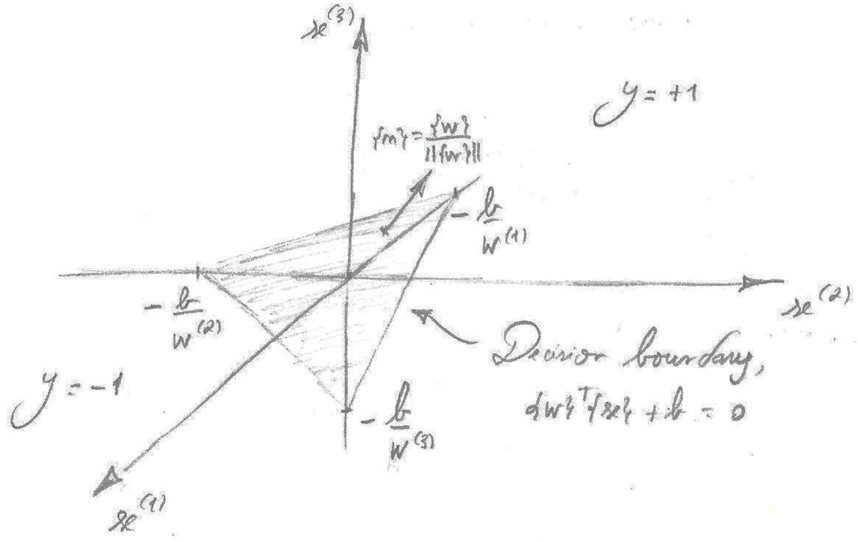
$$\lim_{z\to+\infty} \sigma = 1$$

} The sigmoid function satisfy the probability probability of being non-negative and limited to the range $[0,1]$

$$\frac{d\sigma}{dz} = \frac{e^{-z}}{(1+e^{-z})^2} > 0 \Rightarrow$$ The sigmoid function strictly increases with the argument z

From (459), one understands that the decision boundary in the $\{x\}$ space corresponds to $z = \{w\}^T\{x\} + b = 0$. Equation $\{w\}^T\{x\} + b = 0$ defines a hyperplane of normal $\{m\} = \frac{\{w\}}{\|\{w\}\|}$, as given by (4), that crosses the j-th axis at $-\frac{b}{w^{(j)}}$ (see page 24).

Let $\{x_0\}$ pertain to the decision boundary, therefore, satisfying $\{w\}^T\{x_0\} + b = 0$. One has $\{w\}^T(\{x_0\} + \{\varepsilon\}) + b = \{w\}^T\{\varepsilon\} > 0$ if $\{w\}$ and $\{\varepsilon\}$ make an angle less than $\frac{\pi}{2}$. Therefore, if $\{x\} = \{x_0\} + \{\varepsilon\}$ falls on the side where $\{w\}$ points to, it is labelled with $y = +1$. Otherwise, it is labelled with $y = -1$.



The further the point $\{x\}$ is from the decision boundary (in terms of the normal component of the relative position of the point with respect to the hyperplane) the higher the probability of $\{x\}$ having label $y=1$ if it falls on the side that $\{w\}$ points to or $y=-1$ if it falls on the other side.

As done in the case of Linear Regression (see page 211) one may transform $\{w\}$ and $\{x\}$ so that (457) may be re-expressed as

$$\boxed{P(y=1) = \sigma(\{w\}^T\{x\}) = \frac{1}{1 + e^{-\{w\}^T\{x\}}}} \quad \text{(459)}$$

(Logistic function of $z = \{w\}^T\{x\}$)

The log-likelihood function associated with the dataset $D = \{(\{x_1\}, y_1), \dots (\{x_m\}, y_m)\}$ corresponds to

$$\log L = \log(P(D|\{w\})) = \log\left(P(\{x_1\}, y_1, \dots \{x_m\}, y_m | \{w\})\right) =$$

$$\overset{(\{x_1\},y_1),\dots,(\{x_m\},y_m) \text{ are independent}}{=} \log\left(\prod_{i=1}^{m} P(\{x_i\}, y_i | \{w\})\right) = \sum_{i=1}^{m} \log P(\{x_i\}, y_i | \{w\}) =$$

$$\overset{(52)}{=} \sum_{i=1}^{m} \log\left(P(y_i | \{x_i\}, \{w\}) \cdot P(\{x_i\})\right) =$$

$$= \sum_{i=1}^{m} \log(P(\{x_i\})) + \sum_{i=1}^{m} \log P(y_i | \{x_i\}, \{w\}) \Rightarrow$$

$$\left(P(y_i | \{x_i\}, \{w\}) = \begin{cases} P(y_i = -1 | \{x_i\}, \{w\}) & \text{if } y_i = -1 \\ P(y_i = +1 | \{x_i\}, \{w\}) & \text{if } y_i = +1 \end{cases} \overset{(459)}{=} \right.$$

$$\left. = \sigma(\{w\}^T\{x_i\})^{\frac{1+y_i}{2}} \cdot (1 - \sigma(\{w\}^T\{x_i\}))^{\frac{1-y_i}{2}} \right)$$

$$\log L = \sum_{i=1}^{m} \log(P(\{x_i\})) + \sum_{i=1}^{m} \frac{1+y_i}{2} \log\left(\sigma(\{w\}^T\{x_i\})\right) + \frac{1-y_i}{2} \log\left(1 - \sigma(\{w\}^T\{x_i\})\right) \Rightarrow$$

$$\Rightarrow \log L = \sum_{i=1}^{m} \log P(\{x_i\}) + \sum_{i=1}^{m} -\frac{1+y_i}{2} \log\left(1+e^{-\{w\}^T\{x_i\}}\right)$$

$$+\frac{1-y_i}{2}\left(-\{w\}^T\{x_i\} - \log\left(1+e^{-\{w\}^T\{x_i\}}\right)\right) =$$

(459)

$$= \sum_{i=1}^{m} \log P(\{x_i\}) + \sum_{i=1}^{m} -\frac{1-y_i}{2}\{w\}^T\{x_i\} - \underbrace{\left(\frac{1+y_i}{2}+\frac{1-y_i}{2}\right)}_{=1} \log\left(1+e^{-\{w\}^T\{x_i\}}\right) \Rightarrow$$

$$\Rightarrow \boxed{\log L = \sum_{i=1}^{m} \log P(\{x_i\}) - \sum_{i=1}^{m} \frac{1-y_i}{2}\{w\}^T\{x_i\} + \log\left(1+e^{-\{w\}^T\{x_i\}}\right)} \quad (460)$$

According to the Principle of Maximum Likelihood (G1), the optimum weight vector corresponds to:

$$\{w^*\} = \arg\max_{\{w\}} \log L \Rightarrow$$

$$\Rightarrow \boxed{\{w^*\} = \arg\min_{\{w\}} \underbrace{\sum_{i=1}^{m} \frac{1-y_i}{2}\{w\}^T\{x_i\} + \log\left(1+e^{-\{w\}^T\{x_i\}}\right)}_{=L}} \quad (461)$$

(Logistic Regression problem)

Unfortunately it is not possible to explicitly express the solution of (461). But let us write here the gradient and Hessian matrix of L:

$$\frac{\partial L}{\partial w^{(j)}} = \sum_{i=1}^{m} \frac{1-y_i}{2} x_i^{(j)} - \frac{e^{-\{w\}^T\{x_i\}}}{1+e^{-\{w\}^T\{x_i\}}} x_i^{(j)} \Rightarrow$$

$$\Rightarrow \boxed{\frac{\partial L}{\partial w^{(j)}} = \sum_{i=1}^{m} \left(\frac{1-y_i}{2} - \frac{e^{-\{w\}^T\{x_i\}}}{1+e^{-\{w\}^T\{x_i\}}}\right) x_i^{(j)}} \quad (462)$$

$$\frac{\partial^2 L}{\partial w^{(j)} \partial w^{(l)}} = \sum_{i=1}^{m} \frac{e^{-\{w\}^T\{x_i\}} x_i^{(l)}\left(1+e^{-\{w\}^T\{x_i\}}\right) - e^{-\{w\}^T\{x_i\}} e^{-\{w\}^T\{x_i\}} x_i^{(l)}}{\left(1+e^{-\{w\}^T\{x_i\}}\right)^2} x_i^{(j)} =$$

$$= \sum_{i=1}^{m} \frac{e^{-\{w\}^T\{x_i\}}\left(1+e^{-\{w\}^T\{x_i\}} - e^{-\{w\}^T\{x_i\}}\right)}{\left(1+e^{-\{w\}^T\{x_i\}}\right)^2} x_i^{(l)} x_i^{(j)} \Rightarrow$$

$$\Rightarrow \boxed{\frac{\partial^2 L}{\partial w^{(j)} \partial w^{(l)}} = \sum_{i=1}^{m} \frac{e^{-\{w\}^T\{x_i\}}}{\left(1+e^{-\{w\}^T\{x_i\}}\right)^2} x_i^{(j)} x_i^{(l)}} \quad (463)$$

Note that one may linearize the Logistic Regression problem since as in Linear Regression the dependence on $\{w\}^T\{x\}$ implies that $\{w\}$ is a linear combination of the feature vectors $\{x_i\}$ and that it may be expressed as

$$\{w\} = [A]\{\alpha\}$$

where $[A] = [\{x_1\} \dots \{x_m\}]$ and $\{\alpha\}$ is a vector of constants. See page 227 for more details.

Multiclass Logistic Regression

In the case of y pertaining to a set of more than two elements the components of the softmax function may be used as models for the probability of y corresponding to one of such values.

Let these values be $\alpha_1, \dots, \alpha_K$. The j-th component of the softmax function corresponds to

$$\boxed{P(y=\alpha_j) = \sigma_j(\{w\}^T\{x\}) = \frac{e^{\{w_j\}^T\{x\}}}{\sum_{l=1}^{K} e^{\{w_l\}^T\{x\}}}} \quad (464)$$

(j-th softmax function of $\{z\} = [w]^T\{x\}$)

where $\{w_j\}$ is the j-th weight vector (there are now K weight vectors: one for each class).

The components of the softmax function have the probability properties of non-negativity and normalization:

$$\boxed{\sigma_j > 0} \quad (465) \qquad \boxed{\sum_{j=1}^{K} \sigma_j = 1} \quad (466)$$

Indeed,

$$\sum_{j=1}^{K} \sigma_j \overset{(466)}{=} \sum_{j=1}^{K} \frac{e^{\{w_j\}^T\{x\}}}{\sum_{l=1}^{K} e^{\{w_l\}^T\{x\}}} = \frac{\sum_{j=1}^{K} e^{\{w_j\}^T\{x\}}}{\sum_{l=1}^{K} e^{\{w_l\}^T\{x\}}} = 1$$

From (464) one may understand that σ_j may be expressed through $K-1$ transformed weights instead of the K original ones:

$$\sigma_j = \frac{e^{\{w_j\}^T\{x\}}}{\sum_{l=1}^{K} e^{\{w_l\}^T\{x\}}} \Leftrightarrow \boxed{\sigma_j([w]^T\{x\}) = f([\tilde{w}]^T\{x\}) = \frac{e^{-\{\tilde{w}_j\}^T\{x\}}}{1 + \sum_{\substack{l=1 \\ l \neq s}}^{K} e^{-\{\tilde{w}_l\}^T\{x\}}}}, \quad (467)$$

(Multiplying the numerator and the denominator by $e^{-\{w_s\}^T\{x\}}$)

where the transformed weight vectors are $\boxed{\{\tilde{w}_l\} = \{w_s\} - \{w_l\}}$ (468).

From (467) one then concludes that the softmax function is a generalization of the logistic function for any $K \geq 1$.

Indeed if $K=2$, one of the components of the softmax function coincides with the sigmoid function:

$$\sigma_2([w]^T\{x\}) \overset{(\text{Let } s=2)}{=} \frac{1}{1 + e^{-(\{w_2\}-\{w_1\})^T\{x\}}} \overset{(\text{Let } \{w\} = \{w_2\} - \{w_1\})}{=} \frac{1}{1 + e^{-\{w\}^T\{x\}}} =$$

$$= \sigma(\{w\}^T\{x\})$$

$$\sigma_1([w]^T\{x\}) = \frac{e^{-(\{w_2\}-\{w_1\})^T\{x\}}}{1 + e^{-(\{w_2\}-\{w_1\})^T\{x\}}} = \frac{1 + e^{-(\{w_2\}-\{w_1\})^T\{x\}} - 1}{1 + e^{-(\{w_2\}-\{w_1\})^T\{x\}}} =$$

$$= 1 - \frac{1}{1 + e^{-\{w\}^T\{x\}}} = 1 - \sigma(\{w\}^T\{x\})$$

For the second case, the prediction of Logistic Regression for the label y associated with the feature $\{x\}$ is the value α_j in which the index j is the one that maximizes $\nabla_j([w]^T\{x\})$, that is,

$$y(\{x\}) = \alpha_{j^*} : \; j^* = \underset{j}{\operatorname{argmax}}\; P(y=\alpha_j) = \underset{j}{\operatorname{argmax}}\; \nabla_j([w]^T\{x\}) \tag{469}$$

(Prediction of the Multiclass Logistic Regression)

Note: (Etymology of "softmax function")

The term "softmax function" comes from its contrast with the so-called "hardmax function" - also termed "one-hot argmax function". The "hardmax function" of a vector $\{z\}$ gives the index of the component of $\{z\}$ as a one-hot vector, that is,

$$\{\nabla(\{z\})\} = \{e_{j^*}\}, \text{ where } j^* = \underset{j}{\operatorname{argmax}}\; z^{(j)} \tag{470}$$

("Hardmax function")

The term "one-hot" comes from the fact that "1." (the "hot value") is given to the component of $\{z\}$ for which the component of $\{z\}$ is the maximum one while "0" (the "cold value") is given to all the other components.

→

← The softmax function of a vector $\{z\}$ is a "smoother" version of the hardmax function. The resultant vector $\{\nabla(\{z\})\}$ has still its maximum component in the same place as $\{z\}$, but the values of the components of $\{\nabla(\{z\})\}$ differ less abruptly between themselves than in the case of the hardmax function $\{\nabla(\{z\})\}$.

$$\nabla_j(\{z\}) = \frac{e^{z^{(j)}}}{\sum_{l=1}^{K} e^{z^{(l)}}} \Rightarrow \text{the greater the } z^{(j)} \text{ value the greater the } \nabla_j(\{z\}) \text{ one.}$$

Note that in both softmax and hardmax function the vector $\{z\} \in \mathbb{R}^K$ is mapped into $[0,1]^K$.

XI - Bias - Variance Tradeoff

Let one consider the case of learning function $f(\{x\})$ - the actual law between $\{x\}$ and the "true" y - from S datasets $\{\mathcal{D}_s\}_{s=1}^{S}$, that is, $\{\mathcal{D}_1, \dots, \mathcal{D}_S\}$ where $\mathcal{D}_s = \{(\{x_{s1}\}, y_{s1}), \dots, (\{x_{sm}\}, y_{sm})\}$ using the predictor (model) $\hat{f}(\{x\})$. Let $\hat{f}_s(\{x\})$ be the predictor obtained from dataset $\mathcal{D}_s$, evaluated at $\{x\}$. Also, let

$$\underbrace{y = f(\{x\}) + \varepsilon}_{\text{(Extracted } y\text{)}} \qquad \text{("True" } y\text{)} \tag{471}$$

be the law between $\{x\}$ and extracted y, where ε is some noise of null mean $\boxed{\mathbb{E}(\varepsilon) = 0}$ (472) and some variance $\boxed{Var(\varepsilon) = \sigma^2}$ (473)

Law (471) means that the data points (which are extracted) have label values y that might not be the "true" ones - they are subject to noise.

Bias of the predictor

The bias of the predictor $\hat{f}$ at $\{x\}$ corresponds to the expected value (in terms of the dataset $\mathcal{D}$) of the difference between the prediction and the "true" value:

$$Bias_{\hat{f}}(\{x\}) := \mathbb{E}_{\mathcal{D}}\left(\hat{f}(\{x\}) - f(\{x\})\right) = \quad \leftarrow (\mathbb{E} \text{ is a linear function})$$

$$= \mathbb{E}_{\mathcal{D}}\left(\hat{f}(\{x\})\right) - \mathbb{E}_{\mathcal{D}}\left(f(\{x\})\right) =$$

$\Leftarrow$ $\mathrm{Bias}_{\hat{f}}(f(x^i)) = \mathbb{E}_{\mathcal{D}}(\hat{f}(f(x^i))) - f(f(x^i)) \Leftarrow$

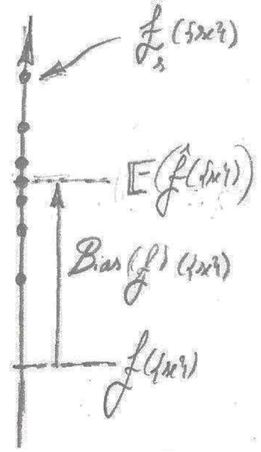
$\mathbb{E}(f(f(x^i))) = f(f(x^i))$

(since $f(f(x^i))$ gives always the same value for $f(x^i)$)

$\Leftarrow$ $\boxed{\mathrm{Bias}_{\hat{f}}(f(x^i)) := \mathbb{E}_{\mathcal{D}}(\hat{f}(f(x^i)) - f(f(x^i))) \approx \left(\frac{1}{S}\sum_{s=1}^{S} \hat{f}_s(f(x^i))\right) - f(f(x^i))}$ (474)

(Bias of predictor $\hat{f}$ at $f(x^i)$)

(Bias is associated with "accuracy")



Variance of the predictor

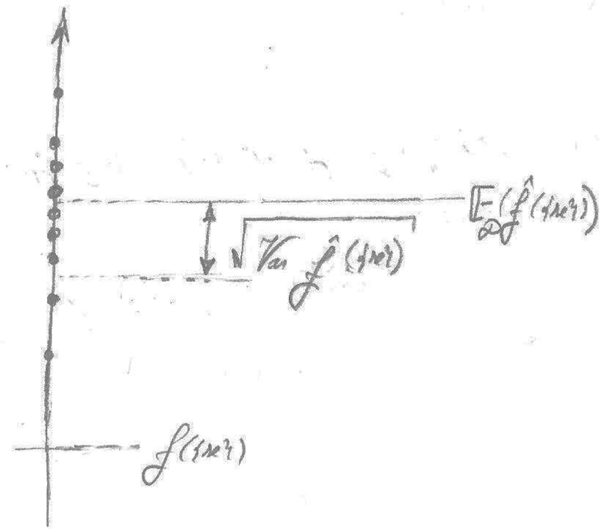
The variance of the predictor $\hat{f}$ at $f(x^i)$ corresponds to the expected value of the squared difference between the prediction and the expected value of the prediction:

$\mathrm{Var}_{\hat{f}}(f(x^i)) := \mathbb{E}_{\mathcal{D}}\left(\hat{f}(f(x^i)) - \mathbb{E}_{\mathcal{D}}(\hat{f}(f(x^i)))\right)^2 \Leftarrow$ (96)

$\Leftarrow$ $\boxed{\mathrm{Var}_{\hat{f}}(f(x^i)) := \mathbb{E}_{\mathcal{D}}\left(\hat{f}(f(x^i)) - \mathbb{E}_{\mathcal{D}}(\hat{f}(f(x^i)))\right)^2 = \frac{1}{S}\sum_{s=1}^{S}\left(\hat{f}_s(f(x^i)) - \frac{1}{S}\sum_{i=1}^{S}\hat{f}_s(f(x^i))\right)^2}$

(Variance of predictor $\hat{f}$ at $f(x^i)$) (475)

(Variance is associated with "precision")



Relation between MSE, bias and variance

The squared error of the predictor $\hat{f}$ at $f(x^i)$ corresponds to

Well, $\boxed{SE(f(x^i)) = (\hat{f}(f(x^i)) - y)^2}$ (476) (Extracted label)

$\mathbb{E}(SE(f(x^i))) = \mathbb{E}_{\mathcal{D}}(\hat{f}(f(x^i)) - y)^2 = \mathbb{E}_{\mathcal{D}}(\hat{f}^2(f(x^i)) - 2y\hat{f}(f(x^i)) + y^2)$

$= \mathbb{E}_{\mathcal{D}}(\hat{f}^2(f(x^i))) - \mathbb{E}_{\mathcal{D}}(2y\hat{f}(f(x^i))) + \mathbb{E}_{\mathcal{D}}(y^2) =$

$\rightarrow = \mathrm{Var}_{\hat{f}}(f(x^i)) + \mathbb{E}_{\mathcal{D}}^2(\hat{f}(f(x^i))) - 2\,\mathbb{E}(y\hat{f}(f(x^i))) + \mathbb{E}_{\mathcal{D}}(y^2) \Leftarrow$

$\left(\mathrm{Var}_{\hat{f}}(f(x^i)) \overset{(475)}{=} \mathbb{E}_{\mathcal{D}}\left(\hat{f}(f(x^i)) - \mathbb{E}_{\mathcal{D}}(\hat{f}(f(x^i)))\right)^2 = \mathbb{E}_{\mathcal{D}}(\hat{f}^2(f(x^i))) - 2\,\mathbb{E}_{\mathcal{D}}(\hat{f}(f(x^i)))\mathbb{E}_{\mathcal{D}}(\hat{f}(f(x^i))) + \mathbb{E}_{\mathcal{D}}^2(\hat{f}(f(x^i))) =\right.$

$\left. = \mathbb{E}_{\mathcal{D}}(\hat{f}^2(f(x^i))) - \mathbb{E}_{\mathcal{D}}^2(\hat{f}(f(x^i)))\right)$

$$\mathbb{E}_{\mathcal{D}}(y\hat{f}(\mathbf{x}_t)) \overset{(471)}{=} \mathbb{E}_{\mathcal{D}}(f(\mathbf{x}_t)\hat{f}(\mathbf{x}_t)) + \mathbb{E}_{\mathcal{D}}(\varepsilon\hat{f}(\mathbf{x}_t)) = f(\mathbf{x}_t)\mathbb{E}_{\mathcal{D}}(\hat{f}(\mathbf{x}_t)) + \mathbb{E}_{\mathcal{D}}(\varepsilon)\mathbb{E}_{\mathcal{D}}(\hat{f}(\mathbf{x}_t)) =$$

(ε and $\hat{f}$ are independent when $n \to +\infty$; $f(\mathbf{x}_t)$ does not depend on $\mathcal{D}$)

$$\overset{(472)}{=} f(\mathbf{x}_t)\mathbb{E}_{\mathcal{D}}(\hat{f}(\mathbf{x}_t))$$

$$\mathbb{E}_{\mathcal{D}}(y^2) \overset{(471)}{=} \mathbb{E}_{\mathcal{D}}(f(\mathbf{x}_t)+\varepsilon)^2 = \mathbb{E}_{\mathcal{D}}(f^2(\mathbf{x}_t)) + \mathbb{E}_{\mathcal{D}}(2f(\mathbf{x}_t)\varepsilon) + \mathbb{E}_{\mathcal{D}}(\varepsilon^2) =$$

$$= f^2(\mathbf{x}_t) + 2f(\mathbf{x}_t)\underbrace{\mathbb{E}(\varepsilon)}_{=0, \text{ from (472)}} + \sigma^2 =$$

($\mathrm{Var}(\varepsilon) = \mathbb{E}(\varepsilon - \underbrace{\mathbb{E}(\varepsilon)}_{=0})^2 = \mathbb{E}(\varepsilon^2) = \sigma^2$)

($f(\mathbf{x}_t)$ is deterministic, not depending on $\mathcal{D}$)

$$= f^2(\mathbf{x}_t) + \sigma^2$$

$$\Leftrightarrow \mathbb{E}_{\mathcal{D}}(\mathcal{TE}(\mathbf{x}_t)) = \mathrm{Var}_{\hat{f}}(\mathbf{x}_t) + \mathbb{E}_{\mathcal{D}}^2(\hat{f}(\mathbf{x}_t)) - 2f(\mathbf{x}_t)\mathbb{E}_{\mathcal{D}}(\hat{f}(\mathbf{x}_t)) +$$

$$+ f^2(\mathbf{x}_t) + \sigma^2 =$$

$$= \mathrm{Var}_{\hat{f}}(\mathbf{x}_t) + \left(f(\mathbf{x}_t) - \mathbb{E}_{\mathcal{D}}(\hat{f}(\mathbf{x}_t))\right)^2 + \sigma^2 \Leftrightarrow$$

($f(\mathbf{x}_t)$ is deterministic)

$$= \mathrm{Var}_{\hat{f}}(\mathbf{x}_t) + \underbrace{\mathbb{E}_{\mathcal{D}}^2\left(\hat{f}(\mathbf{x}_t) - f(\mathbf{x}_t)\right)}_{= \mathrm{Bias}^2_{\hat{f}}(\mathbf{x}_t), \text{ from (474)}} + \sigma^2 \Leftrightarrow$$

$$\Leftrightarrow \boxed{\mathbb{E}_{\mathcal{D}}(\mathcal{TE}(\mathbf{x}_t)) = \mathbb{E}_{\mathcal{D}}\left(\hat{f}(\mathbf{x}_t) - y\right)^2 = \mathrm{Bias}^2_{\hat{f}}(\mathbf{x}_t) + \mathrm{Var}_{\hat{f}}(\mathbf{x}_t) + \sigma^2}$$

(Expected value (in terms of the datasets $\mathcal{D}$) of the squared error at $\mathbf{x}_t$) (477)

The mean squared error of the predictor $\hat{f}$ is the expected value in both the feature value $\mathbf{x}_t$ and dataset $\mathcal{D}$ of the squared error, that is,

$$MSE := \mathbb{E}_{\mathbf{x}_t}\left(\mathbb{E}_{\mathcal{D}}(\mathcal{TE}(\mathbf{x}_t))\right) \overset{(477)}{=}$$

$$= \mathbb{E}_{\mathbf{x}_t}\left(\mathrm{Bias}^2_{\hat{f}}(\mathbf{x}_t)\right) + \mathbb{E}_{\mathbf{x}_t}\left(\mathrm{Var}_{\hat{f}}(\mathbf{x}_t)\right) + \mathbb{E}_{\mathbf{x}_t}(\sigma^2)_{(\varepsilon)}$$

Expected squared bias — Expected variance — Label's noise

$$\Leftrightarrow \boxed{MSE = \mathbb{E}_{\mathbf{x}_t}\left(\mathrm{Bias}^2_{\hat{f}}(\mathbf{x}_t)\right) + \mathbb{E}_{\mathbf{x}_t}\left(\mathrm{Var}_{\hat{f}}(\mathbf{x}_t)\right) + \sigma^2} \quad (478)$$

Note: (Differences between (478) and the MSE obtained from solely training points and a test dataset $\mathcal{D}$)

One should be careful with the definition of (478) MSE. Usually the MSE for a test dataset $\mathcal{D}_s$ considering $\mathbf{x}_t$ to coincide with the training values is defined as

$$MSE^{train}_{\mathcal{D}_s} = \sum_{\mathbf{x}_{s,i}, y_{s,i} \in \mathcal{D}_s} \left(\hat{f}_s(\mathbf{x}_{s,i}) - y_{s,i}\right)$$

Herein $\mathbf{x}_t$ do not correspond to any point in the whole domain except to the training values. In (478) $\mathbf{x}_t$ may take any value regardless of the applied dataset. →

Therefore,

$$MSE \neq \sum_{s=1}^{S} \sum_{i=1}^{n} \left(f_s(x_{s,i}) - y_{s,i} \right).$$

From (478), one understands that the MSE is sum of three terms: one due to bias, another due to variance and another due to noise. All of them are non-negative. The latter is irreducible! There is usually a trade-off between bias and variance: a reduction on bias increases an increase of variance and vice-versa.

Overfitting usually increases the variance: in this case an overly complex model is fitted to the training data that could even "capture" its noise. The resultant curve would change abruptly from training set to training set and keeping a low training MSE. However the overall MSE defined by (478) might be high since a curve that captures almost all the structure of a training set might not capture the testing one...

Underfitting usually increases bias: in this case an overly simple model is fitted to the training data that has just some structure of the data. Both the training MSE and overall MSE might be significant. Still, usually, due to its simplicity, the curve does not vary too much from training set to training set, and the overall MSE might not differ so much from the training one.

Weak and strong learners

A weak learner is one that performs relatively poorly - its accuracy is above chance, but just barely, and has a low variance. An example of a weak learner is a decision tree stump (a decision tree with just one split - or may be two). Weak learners might be combined together through "ensemble learning" to obtain a strong learner, which has a much greater accuracy while maintaining a low variance. Examples of ensemble learning methods are "boosting" and "bagging".

"Vanilla" bagging algorithm

"Bagging" comes from "Bootstrap Aggregating". The "bootstrap" term comes from the process of picking S random samples $\mathcal{D}_s$ ($s = 1, \dots, S$) from a set sample $\mathcal{D}$ through sampling with replacement - it may be viewed as a way of "pulling yourself from your own boot straps" as it is a process that does not add any new information (it uses what it already got). The S samples are named "bootstrap samples" or "bootstraps". Each sample has the same number of elements as the original one, n. Let one describe such samples.

The probability of the i-th point of $\mathcal{D}$, $(\{x_m\}, y_m)$ belonging to the created bootstrap sample $\mathcal{D}_1$ corresponds to

$$P\left((\{x_m\}, y_m) \in \mathcal{D}_1\right) = 1 - P\left((\{x_m\}, y_m) \notin \mathcal{D}_1\right) =$$

$$= 1 - P\left((\{x_{1,1}\}, y_{1,1}) \neq (\{x_m\}, y_m), \dots, (\{x_{m,m}\}, y_{1,m}) \neq (\{x_m\}, y_m)\right) =$$

$$= 1 - \prod_{i=1}^{m} P\left((\{x_{1,i}\}, y_{1,i}) \neq (\{x_m\}, y_m)\right)$$

(Each drawing with replacement from each other)

$$= 1 - \prod_{i=1}^{m} \left(1 - P\left((\{x_{1,i}\}, y_{1,i}) = (\{x_m\}, y_m)\right)\right) =$$

$$= \frac{1}{m}$$

$$= 1 - \prod_{i=1}^{m} \left(1 - \frac{1}{m}\right) = 1 - \left(1 - \frac{1}{m}\right)^m =$$

$$\overset{(39)}{=} 1 - \sum_{k=0}^{m} \frac{m!}{k!(m-k)!} \left(-\frac{1}{m}\right)^k$$

$$= 1 - \sum_{k=0}^{m} \frac{(-1)^k}{k!} \frac{m!}{(m-k)!\, m^k}$$

$$\left(\frac{m!}{m!\, m^0} = 1, \quad \frac{m!}{(m-1)!\, m^1} = \frac{m(m-1)!}{(m-1)!\, m} = 1, \quad \frac{m!}{(m-2)!\, m^2} = \frac{m(m-1)(m-2)!}{(m-2)!\, m^2} = \frac{m^2 - m}{m^2} \xrightarrow[m \to +\infty]{} 1 \right.$$

$$\left. \frac{m!}{(m-k)!\, m^k} \xrightarrow[m \to +\infty]{} 1 \right)$$

$$\xrightarrow[m \to +\infty]{} 1 - \sum_{k=0}^{m} \frac{(-1)^k}{k!} \quad (=)$$

$$\underbrace{\sum_{k=0}^{m} \frac{(-1)^k}{k!}}_{\xrightarrow[m \to +\infty]{} e^{-1}}$$

$$\boxed{P\left((\{x_m\}, y_m) \in \mathcal{D}_1\right) \xrightarrow[m \to +\infty]{} 1 - e^{-1} \simeq 0.632} \tag{479}$$

(Probability of point $(\{x_m\}, y_m)$ belonging to the bootstrap sample $\mathcal{D}_1$)

Let $\tilde{m}$ be the number of unique elements of $\mathcal{D}$ appearing in $\mathcal{D}_1$. The expected value for $\tilde{m}$ is

$$\boxed{\mathbb{E}(\tilde{m}) \xrightarrow[m \to +\infty]{} m(1 - e^{-1}) \simeq 0.632\, m} \tag{480}$$

(Expected number of unique elements of $\mathcal{D}$ appearing in the bootstrap sample $\mathcal{D}_1$)

One may reach result (480) through intuition.

Let $f(\{x\})$ be the weak learner. In bagging one gets S trained weak learners by separately using the data of the bootstrap samples, $\mathcal{D}_1, \dots, \mathcal{D}_S$. Let f_s be the weak learner obtained from the bootstrap sample $\mathcal{D}_s$.

The "aggregating" term of the algorithm comes from the fact that the strong learner is defined so that the predicted label y for $\{x\}$ corresponds to the arithmetic mean of the weak learners in the case of regression, that is,

$$\boxed{y = f(\{x\}) := \frac{1}{S} \sum_{s=1}^{S} f_s(\{x\})} \tag{481}$$

(Prediction of the bagging algorithm for the case of regression)

or the mode of the weak learners in the case of classification, that is,

$$\boxed{y = f(\{x\}) = \text{mode}\left(\{f_s(\{x\})\}_{s=1}^{S}\right)} \tag{482}$$

(Prediction of the bagging algorithm for the case of classification)

The main motivation for using bagging is that a model with a lower variance is obtained. And this occurs while not (virtually) changing the bias.

Such reduction may be achieved by understanding the effect of averaging on the mean and variance of a quantity:

$$\mu = \mathbb{E}\left(\frac{1}{S}\sum_{s=1}^{S} f_s\right) = \frac{1}{S}\sum_{s=1}^{S}\mathbb{E}(f_s) = \frac{S}{S}\mu_s(=)\boxed{\mu = \mu_s} \quad (483)$$

(f_s are identically distributed. Therefore $\mathbb{E}(f_s) = \mu_s$ is the same for all s)

$$\sigma^2 = \mathrm{Var}\left(\frac{1}{S}\sum_{s=1}^{S} f_s\right) \overset{(104)}{=} \frac{1}{S}\underbrace{\mathrm{Var}(f_s)}_{=:\sigma_s^2}(=)\boxed{\sigma^2 = \frac{1}{S}\sigma_s} \quad (484)$$

Random Forest algorithm

Random Forest is a type of bagging algorithm applied to decision trees, hence the term "Forest". The Random Forest algorithm consists of:

For $s = 1, \dots, S$:

1- Create a bootstrap sample $\mathcal{D}_s$ from the training data $\mathcal{D}$.

2- Train a decision tree f_s using the data from the bootstrap sample $\mathcal{D}_s$. Such training is done iteratively by repeating the following steps until some stopping criteria are met: →

←

2.1 - Randomly select m components of the p from the D ones.

2.2 - Pick the index j from the m ones and the threshold t whose decision criterion $x^{(j)} < t$ has the highest information gain.

2.3 - Split the node into two child ones, using the condition found in 2.2.

The prediction y for the feature $x \in \mathbb{R}^p$ is done as in the case of "vanilla" bagging (see (481) and (482)).

The sole differences between the Random Forest algorithm and the "vanilla" bagging one is that the former uses solely decision trees and the fact that there's an additional step (step 2.1) in training that makes the decision trees to be less correlated. Indeed, the reduction of the variance as described by equation (484) only holds if the weak learners f_s are uncorrelated (see Bienaymé's identity (104)).

Correlated trees would be more identical and their aggregation would be similar to a sole tree which would lead the ensemble learning to be useless.

The Random Forest algorithm has two hyper-parameters: the number of bootstrap samples, S, and the number of components of $x \in \mathbb{R}^p$, m, to consider in the splitting process when training the decision trees. Typically S is in the order of the hundreds. The creators of the algorithm recommend the following values for m:

$$\boxed{m = \begin{cases} \sqrt{p}, & \text{if classification} \\ p/3, & \text{if regression} \end{cases}} \quad (485)$$

The main motivation for using bagging is really to reduce the variance. Since the bias is usually unaffected the weak learners should have low bias. Decision trees usually have low bias a but high variance which makes the Random Forest algorithm to be highly desirable.

Boosting

Boosting consists of using the original training data D and iteratively creating multiple models by using a weak learner. Each new model would be different from the previous ones in the sense that the weak learner, by building each new model tries to "fix" the errors which previous models make.

The final ensemble model is a certain combination of those multiple weak models built iteratively.

Let us now consider that the current problem is binary classification with $y \in \{-1, 1\}$. The ensemble model in Boosting corresponds to a basis function expansion:

$$y = \operatorname{sgn}(f(\{x\})) = \operatorname{sgn}\left(\sum_{m=1}^{M} \beta_m g(\{x\}; \{\theta_m\})\right) \tag{486}$$

(Boosting's binary classification model)

where $\beta_1, \dots, \beta_M$ are the expansion coefficients and $g(\{x\}, \{\theta_m\})$ is the basis function of argument $\{x\}$ and described by the vector of parameters $\{\theta_m\}$. The base function g is also a binary classifier returning -1 or $+1$.

Boosting regards the so-called Forward Stagewise Additive Modelling which successively constructs functions

$f_m(\{x\}) = f_{m-1}(\{x\}) + \beta_m g(\{x\}; \{\theta_m\})$ whose parameters β_m and $\{\theta_m\}$ are the ones that minimize some cost function. While finding those parameters the ones of $f_{m-1}(\{x\})$ are unchanged.

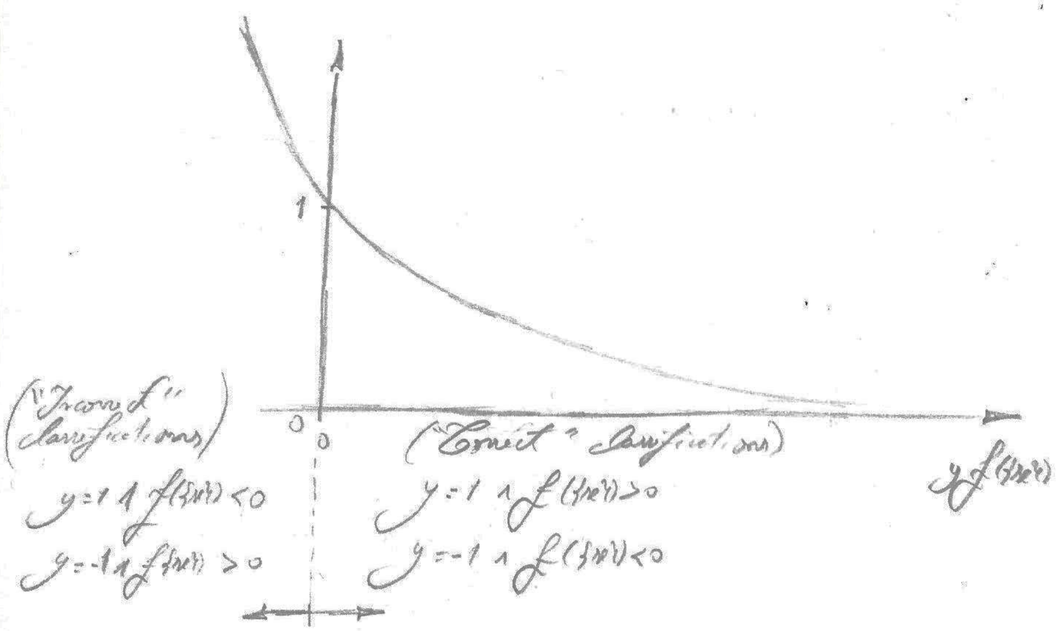
The Forward Stagewise Additive Modelling is defined as

1. Initialize $f_0(\{x\}) = 0$.
2. For $m = 1, \dots, M$:
   2.1. Compute
   $$(\beta_m, \{\theta_m\}) = \underset{(\beta, \{\theta\})}{\operatorname{argmin}} \sum_{i=1}^{n} L\left(y_i, f_{m-1}(\{x_i\}) + \beta g(\{x_i\}; \{\theta\})\right) \tag{487}$$
   where $L(y, f(\{x\}))$ is some user-defined loss function,
   2.2. Set
   $$f_m(\{x\}) = f_{m-1}(\{x\}) + \beta_m g(\{x\}, \{\theta_m\}) \tag{488}$$

Binary AdaBoost algorithm

The binary AdaBoost algorithm is obtained by choosing the loss function $L(y, f(x))$ to be the so-called exponential loss function:

$$L(y, f(x)) = e^{-y f(x)} \tag{489}$$



From the picture above one notices that the exponential loss function penalizes more false positives or negatives than true positives or negatives as any loss function should.

Let us now one set β_m and $g_m = g(x; \theta_m)$ using (489):

$$(\beta_m, g_m) = \underset{(\beta, g)}{\operatorname{argmin}} \sum_{i=1}^{n} e^{-y_i \left(f_{m-1}(x_i) + \beta g(x_i) \right)} \quad (\Leftrightarrow)$$

$$(\Leftrightarrow) \boxed{(\beta_m, g_m) = \underset{(\beta, g)}{\operatorname{argmin}} \sum_{i=1}^{n} w_{m,i} \cdot e^{-\beta y_i g(x_i)}}, \tag{490}$$

where $\boxed{w_{m,i} = e^{-y_i f_{m-1}(x_i)}}$ (491) Note that $w_{m,i}$ does not depend on β and g and may therefore be regarded as a "weight" for the contribution of the added term, $\beta g(x_i) = f_m(x_i) - f_{m-1}(x_i)$, to the loss function evaluated at x_i, y_i. However, $w_{m,i}$ depends on $f_{m-1}(x_i)$ and, therefore, it is updated from iteration to iteration:

$$w_{m+1,i} = e^{-y_i f_m(x_i)} = e^{-y_i \left(f_{m-1}(x_i) + \beta_m g_m(x_i) \right)}$$

$$= \underbrace{e^{-y_i f_{m-1}(x_i)}}_{= w_{m,i}} \cdot e^{-\beta_m y_i g_m(x_i)} \quad \overset{(488)}{=}$$

(The base function g_m returns a binary value: either -1 or $+1$. Thus $\boxed{I(y_i \neq g_m(x_i)) = \begin{cases} 0 & \text{if } y_i = g_m(x_i) \\ 1 & \text{if } y_i \neq g_m(x_i) \end{cases}}$ (492). One may write

$$-y_i g_m(x_i) = \begin{cases} -1 & \text{if } y_i = g_m(x_i) \\ +1 & \text{if } y_i \neq g_m(x_i) \end{cases} \quad (\Leftrightarrow) \boxed{-y_i g_m(x_i) = 2 I(y_i \neq g_m(x_i)) - 1} \tag{493}$$

$$= w_{m,i} \cdot e^{2\beta_m I(y_i \neq g_m(x_i))} \cdot e^{-\beta_m} \quad (\Leftrightarrow)$$

(Set $\boxed{\alpha_m = 2\beta_m}$ (494))

(The term $e^{-\beta_m}$ is the same for all i and, therefore, it is irrelevant in the minimization process (490))

$$\Leftrightarrow \boxed{w_{m+1,i} = w_{m,i} \cdot e^{\alpha_m I(y_i \neq g_m(x_i))}} \tag{495}$$

Let us find the solution β_m of (490) for fixed β. Well,

$$\sum_{i=1}^{m} w_{m,i} e^{-\beta y_i g(x_i)} = \left(\sum_{\substack{i=1 \\ y_i = g(x_i)}} w_{m,i} e^{-\beta}\right) + \left(\sum_{\substack{i=1 \\ y_i \neq g(x_i)}} w_{m,i} e^{\beta}\right) =$$

$$= e^{-\beta} \sum_{i=1}^{m} w_{m,i} \left(1 - I(y_i \neq g(x_i))\right) + e^{\beta} \sum_{i=1}^{m} w_{m,i} I(y_i \neq g(x_i))$$

$$\Leftrightarrow \boxed{\sum_{i=1}^{m} w_{m,i} e^{-\beta y_i g(x_i)} = \left(e^{\beta} - e^{-\beta}\right) \sum_{i=1}^{m} w_{m,i} I(y_i \neq g(x_i)) + e^{-\beta} \left(\sum_{i=1}^{m} w_{m,i}\right)} \tag{496}$$

Well, if $\beta < 0$, $e^{\beta} < e^{-\beta}$ and $e^{\beta} - e^{-\beta} < 0$. If $\beta > 0$, $e^{\beta} - e^{-\beta} > 0$. Hence

$$\boxed{g_m = \begin{cases} \arg\min_g \sum_{i=1}^{m} w_{m,i} I(y_i \neq g(x_i)) & \text{if } \beta > 0 \\ \arg\max_g \sum_{i=1}^{m} w_{m,i} I(y_i \neq g(x_i)) & \text{if } \beta < 0 \end{cases}} \tag{497}$$

Let

$$\boxed{\varepsilon_m = \frac{\sum_{i=1}^{m} w_{m,i} I(y_i \neq g_m(x_i))}{\sum_{i=1}^{m} w_{m,i}}} \tag{498}$$

be the weighted average of the error of g_m.

By inserting (498) into (496), one obtains

$$\boxed{\sum_{i=1}^{m} w_{m,i} e^{-\beta y_i g_m(x_i)} = \left(\sum_{i=1}^{m} w_{m,i}\right)\underbrace{\left(\left(e^{\beta} - e^{-\beta}\right)\varepsilon_m + e^{-\beta}\right)}_{:= h(\beta)}} \tag{499}$$

Well,

$$\boxed{\frac{dh}{d\beta} = \left(e^{\beta} + e^{-\beta}\right)\varepsilon_m - e^{-\beta}} \tag{500}$$

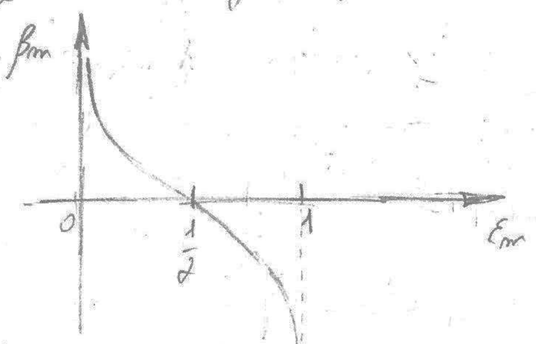
$$\frac{d^2h}{d\beta^2} = \left(e^{\beta} - e^{-\beta}\right)\varepsilon_m + e^{-\beta} = \frac{\sum_{i=1}^{m} w_{m,i} e^{-\beta y_i g_m(x_i)}}{\sum_{i=1}^{m} w_{m,i}} > 0$$

Therefore, h is a convex function of β.

$$\left.\frac{dh}{d\beta}\right|_{\beta_m = 0} = 0 \Leftrightarrow \left(e^{\beta_m} + e^{-\beta_m}\right)\varepsilon_m - e^{-\beta_m} = 0 \Leftrightarrow \left(e^{2\beta_m} + 1\right)\varepsilon_m - 1 = 0 \Leftrightarrow$$

$$\Leftrightarrow \boxed{\beta_m = \frac{1}{2} \log\left(\frac{1 - \varepsilon_m}{\varepsilon_m}\right)} \tag{501}$$

Well, since $I(y_i \neq g_m(x_i)) \in \{0,1\}$, one knows that $\varepsilon_m \in [0,1]$ (ε_m is a weighted average of $I(y_i \neq g_m(x_i))$). Therefore, $\beta_m \geq 0$ if $\varepsilon_m \leq \frac{1}{2}$ and $\beta_m \leq 0$ if $\varepsilon_m \geq \frac{1}{2}$.



If the weak learner $g_m(x_i)$ has reasonable accuracy, then $\varepsilon_m < \frac{1}{2}$ and $\beta_m > 0$, $\alpha_m = 2\beta_m > 0$. The higher the accuracy (that is, the lower the ε_m), the higher the weight β_m of the function g_m in the ensemble function $f(x_i) = \sum_{m=1}^{M} \beta_m g_m(x_i)$. That means that the more accurate weak learners would have a higher weight in the final model. Also, the higher the accuracy, the higher the factor $e^{\alpha_m I(y_i \neq g_m(x_i))}$ to be multiplied by the weight $w_{m,i}$ of an incorrect classification by g_m in order to obtain the weight $w_{m+1,i}$ for the following iteration (see (495)). This means that the following weak learner would be trained to more significantly predict such observation. Increased $w_{m,i}$ weights are said to have been "boosted". And the algorithm is also said to be adaptative due to the adjustment of the weights $w_{m,i}$ according to the accuracy of the previous weak learner on the respective observations.

Note also that from the "proper Bayes" decision principle (59), one has

$$y^* = \underset{y}{\operatorname{argmin}} \mathcal{R}(f|x_i) \overset{(58)}{=} \underset{f(x_i)}{\operatorname{argmin}} \sum_j L(y_j, f(x_i)) P(y_j|x_i) =$$

$$= \underset{f(x_i)}{\operatorname{argmin}} L(y=+1, f(x_i)) P(y=+1|x_i) + L(y=-1, f(x_i)) P(y=-1|x_i) =$$

$$f^*(x_i) = \underset{f(x_i)}{\operatorname{argmin}} \; e^{-f(x_i)} P(y=+1|x_i) + e^{f(x_i)} (1 - P(y=+1|x_i)) \quad (498)$$

$$= h(f) \qquad (502)$$

$$\frac{dh}{df} = -e^{-f(x_i)} P(y=+1|x_i) + e^{f(x_i)} (1 - P(y=+1|x_i))$$

$$\frac{d^2h}{df^2} = e^{-f(x_i)} P(y=+1|x_i) + e^{f(x_i)} (1 - P(y=+1|x_i)) =$$

$$= \mathcal{R}(f|x_i) \geq 0$$

Therefore, h is a convex function on f.

$$\left.\frac{dh}{df}\right|_{f^*} = 0 \Leftrightarrow -e^{-2f^*(x_i)} P(y=+1|x_i) + 1 - P(y=+1|x_i) \Leftrightarrow$$

$$\Leftrightarrow \boxed{P(y=+1|x_i) = \frac{1}{1 + e^{-2f^*(x_i)}}} \qquad (503)$$

From (503) one concludes that $P(y=+1|x_i) > \frac{1}{2}$ if $f^*(x_i) > 0$ and $P(y=-1|x_i) < \frac{1}{2}$ if $f^*(x_i) < 0$. Therefore, one should choose

$$y = \operatorname{sgn}(f(x_i))$$

as described by (486).

The binary AdaBoost algorithm is described by the steps below.

1. Initialize observation weights $w_i = \frac{1}{N}$ for $i = 1, \dots, N$.

2. For $m = 1, \dots, M$:

2.1. Define the weighted average of the errors of $g_m(\{x_i\})$ as

$$\boxed{\varepsilon_m = \frac{\sum_{i=1}^{n} w_i \, I(y_i \neq g_m(\{x_i\}))}{\sum_{i=1}^{n} w_i}} \tag{504}$$

2.2. Find

$$\boxed{g_m = \begin{cases} \underset{g_m}{\operatorname{argmin}}\, \varepsilon_m, & \text{if } \alpha_m \geq 0 \\ \underset{g_m}{\operatorname{argmax}}\, \varepsilon_m, & \text{if } \alpha_m < 0 \end{cases}} \quad \text{and} \quad \boxed{\alpha_m = \log\left(\frac{1-\varepsilon_m}{\varepsilon_m}\right)} \tag{505, 506}$$

such that

$$\boxed{(g_m, \alpha_m) = \underset{(g_m, \alpha_m)}{\operatorname{argmin}} \sum_{i=1}^{n} w_i \, e^{-\frac{1}{2}\alpha_m y_i g_m(\{x_i\})}} \tag{507}$$

2.3. Update weights:

$$\boxed{w_i \leftarrow w_i \, e^{\alpha_m I(y_i \neq g_m(\{x_i\}))} \quad \text{for } i = 1, \dots, n} \tag{508}$$

3. Output

$$\boxed{g(\{x_i\}) = \operatorname{sign}\left(\sum_{m=1}^{M} \alpha_m g_m(\{x_i\})\right)} \tag{509}$$

Gradient Boosting

Gradient Boosting resembles Gradient Descent. If the function L evaluated at the different data points $\{x_i, y_i\}$, that is, $L(\{x_i, y_i\}, f)$, is regarded as a variable, the optimization problem corresponds to find the optimum vector $\{f\} = [L(\{x_1, y_1\}, f), \dots, L(\{x_n, y_n\}, f)]^T$ that minimizes the cost function:

$$\boxed{\{f^*\} = \underset{\{f\}}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, f^{(i)})} \tag{510, 511}$$

By using gradient descent, the vector $\{f^*\}$ is found by considering an initial estimate $\{f_0\}$ and iterating by adding to estimate $\{f_{m-1}\}$ the term $-\beta \{\nabla_{\{f\}} \sum_{i=1}^{n} L\}(y_i, f_{m-1}^{(i)})$, that is,

$$\begin{aligned} \{f_m\} &= \{f_{m-1}\} - \beta \{\nabla_{\{f\}} \sum_{i=1}^{n} L\}(y_i, f_{m-1}^{(i)}) \\ &= \{f_{m-1}\} - \beta \sum_{i=1}^{n} \frac{\partial L}{\partial f^{(i)}}(y_i, f_{m-1}^{(i)}) \{e_i\} \end{aligned}$$

$$\boxed{\{f_m\} = \{f_{m-1}\} - \beta \sum_{i=1}^{n} \left.\frac{\partial L(y_i, f)}{\partial f}\right|_{f = f_{m-1}^{(i)}} \{e_i\}} \tag{512}$$

where β is some non-negative constant, until convergence. Furthermore the constant β would be defined as the one that makes $\{f_m\}$ given by (512) to minimize $\sum_{i=1}^{n} L(y_i, f_m^{(i)})$, that is,

$$\boxed{\beta_m = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} L\left(y_i, f_{m-1}^{(i)} - \beta \left.\frac{\partial L(y_i, f)}{\partial f}\right|_{f = f_{m-1}^{(i)}}\right)} \tag{513}$$

Note that if there was a β for each component of $\{f_m\}$, that is $\beta^{(i)}$, and (513) was minimised for every $\beta^{(i)}$, such minimisation would be equivalent to (511), which would not further simplify the problem. But because there is solely a β for all components of $\{f_m\}$, problem (513) is simpler, although not equivalent, requiring the whole iterative process to achieve the solution of (511).

By treating $f^{(1)} = f(\{x_1, y\}), \dots, f^{(m)} = f(\{x_m, y\})$ as independent variables, one is assuming that function f is infinitely flexible, and equations (512) and (513) would be re-written as

$$\boxed{f_m^{(\{x_i,y\})} = f_{m-1}^{(\{x_i,y\})} - \beta \left.\frac{\partial L(y_i, f)}{\partial f}\right|_{f = f_{m-1}(\{x_i,y\})}} \tag{513}$$

$$\boxed{\beta_m = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{m} L\left(y_i,\ f_{m-1}(\{x_i,y\}) - \beta \left.\frac{\partial L(y_i,f)}{\partial f}\right|_{f = f_{m-1}(\{x_i,y\})}\right)} \tag{514}$$

In Boosting, function $f(\{x,y\})$ is not infinitely flexible. It has some structure. Therefore, the recursion (488) might not satisfy (513). One may however define the weak learner $g_m(\{x,y\})$ as the best approximation to $-\left.\frac{\partial L(y_i,f)}{\partial f}\right|_{f = f_{m-1}(\{x,y\})}$ for all $\{x_i, y\}$.

Let

$$\boxed{r_{m,i} = -\left.\frac{\partial L(y_i,f)}{\partial f}\right|_{f = f_{m-1}(\{x_i,y\})}} \tag{515}$$

be the so-called "pseudo-residual" associated with the m-th iteration and i-th observation. The optimum weak learner $g_m(\{x,y\})$ would be one that best fits to the points $\{(\{x_i,y\}, r_{m,i})\}_{i=1}^{n}$.

Note that one further needs to define the initial estimate $f_0\{x,y\} = [f_0(\{x_1,y\}) \dots f_0(\{x_n,y\})]^T$ from some function $f_0(\{x,y\})$. In regression such function is usually chosen as the constant $f(x) = c$ that minimises the cost:

$$\boxed{f_0(\{x,y\}) = \underset{c}{\operatorname{argmin}} \sum_{i=1}^{m} L(y_i, c)} \tag{516}$$

The gradient Boosting algorithm for regression may be expressed as

1. Initialise with

$$\boxed{f_0(\{x,y\}) = \underset{c}{\operatorname{argmin}} \sum_{i=1}^{m} L(y_i, c)} \tag{517}$$

where $L(y_i, f(\{x_i,y\}))$ is the loss function associated with the i-th observation and $f(\{x,y\}) = c$ is the constant function.

2. For $m = 1, \dots, M$:

2.1. Compute the "pseudo-residuals":

$$r_{im} = -\left.\frac{\partial L(y_i, f)}{\partial f}\right|_{f=f_{m-1}(x_i)} \quad \text{for } i=1,\dots,n \tag{518}$$

2.2. Find the weak learner $g_m(x)$ that best fits into the points $\{(x_i, r_{im})\}_{i=1}^{n}$.

2.3. Compute the optimum learning rate:

$$\beta_m = \arg\min_{\beta} \sum_{i=1}^{n} L\left(y_i, f_{m-1}(x_i) + \beta g_m(x_i)\right) \tag{519}$$

2.4. Update the model:

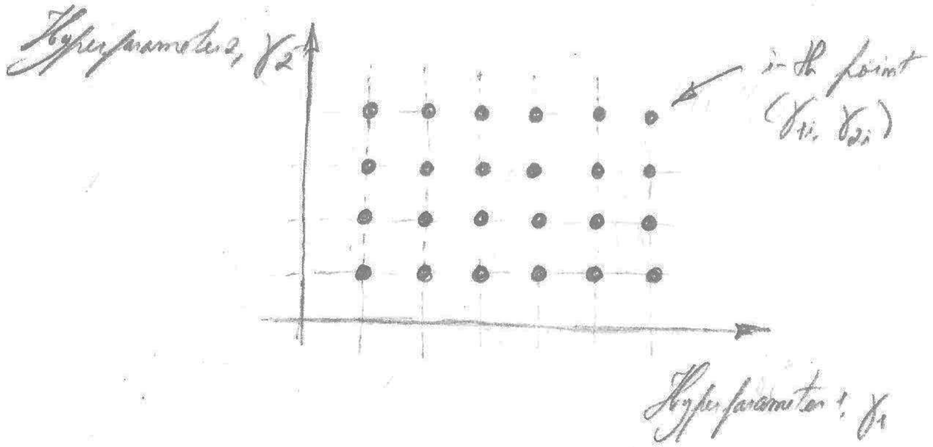
$$f_m(x) = f_{m-1}(x) + \beta_m g_m(x) \tag{520}$$

XII - Model Performance Assessment

The three sets

The data points that are obtained from the experiments should be shuffled and divided into three groups:

1. Training set (70-90%) - it is simply used for training the model.
2. Validation set (15-25%) - this is a "holdout set" (that is, to be put on hold when training the model). It is used for tuning the hyperparameters of the model. The simplest hyperparameter tuning technique is "grid search". The values to be tried follow a "cartesian distribution":

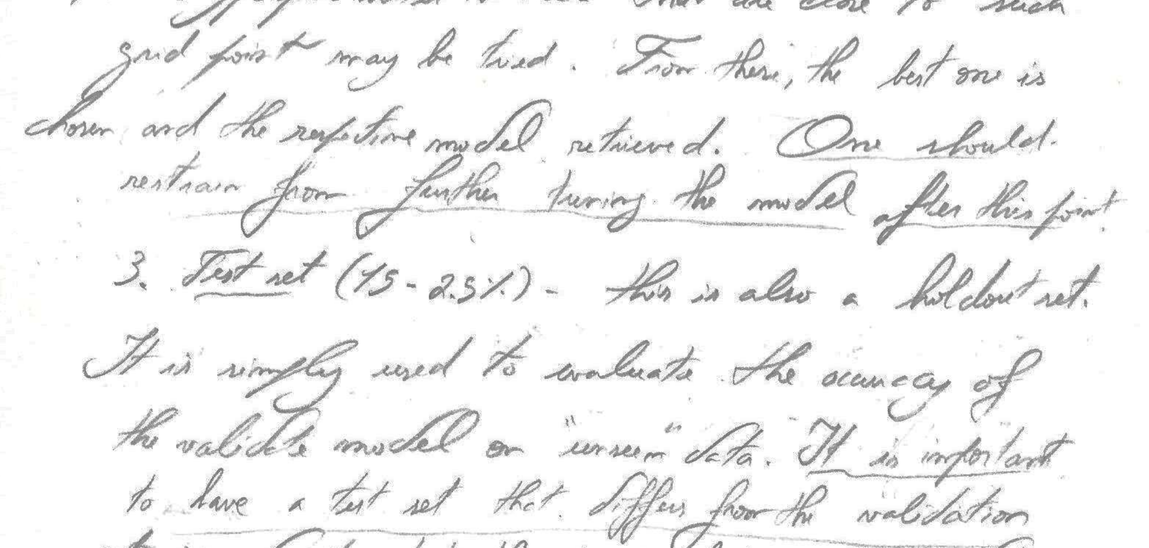


The model is re-trained using the training data and the hyperparameter values for each grid point. The grid point that gives the smallest error associated with the prediction of the validation data is picked and a fine tuning may be performed: that

is, new hyperparameter values that are close to such grid point may be tried. From there, the best one is chosen and the respective model retrieved. One should restrain from further tuning the model after this point.

3. Test set (15 - 25%) - this is also a holdout set. It is simply used to evaluate the accuracy of the validated model on "unseen" data. It is important to have a test set that differs from the validation set in order to test the tuned hyperparameter or data that was not used in its tuning.

The whole process of obtaining the model may be summarized as in the following diagram:



Cross-validation

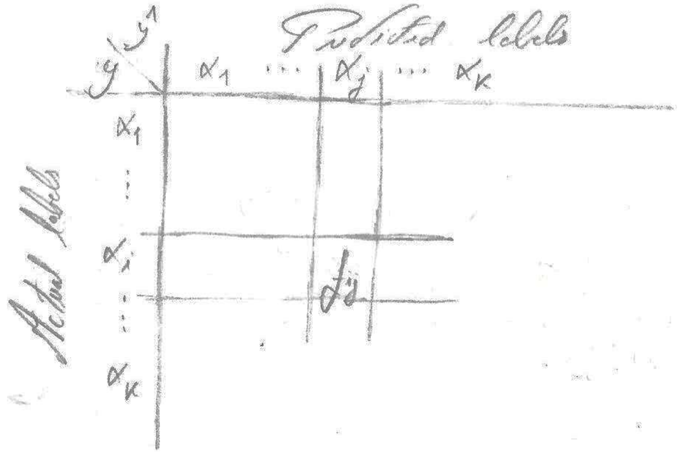
If there is not enough data to build both validation and test sets, one could consider instead solely two sets: the training and the testing sets in which the training one is used for both training and validation, through a technique called "cross-validation". One of the most widely used types of cross-validation is the so-called "K-fold cross validation". This is such that the training data is divided into K sets ("folds") of equal size. To ensure randomness in the K sets, the data points of the original set $\mathcal{D}$ should be shuffled before being divided. From the K datasets, K models are created: the j-th model, $f_j^{(\{\gamma\},\{\theta\})}$ is trained using all datasets except the j-th one. The j-th model is then validated against the data of the j-th set and optimum hyperparameters are obtained. One may then choose from the k sets of optimum hyperparameters the best of all and use it to at last retrain the model using the whole original training set and testing it against the holdout test set.

Usually one chooses $K = 10$.

Classification performance metrics

Confusion matrix

A confusion matrix is such that its (i,j)-th entry, f_{ij}, corresponds to the occurrence frequency of predictions of the j-th class when the i-th class was the actual (correct) one. The rows are then associated with the actual label, y, and the columns with the predicted label $\hat{y}$. An accurate model would have much higher values at the diagonal than at the other entries.



In the table above $\alpha_1, \dots, \alpha_K$ are the values the label y may take. The frequency f_{ij} is defined as

$$f_{ij} = \left| \left\{ y = \alpha_i \wedge \hat{y}(\{x_i\}) = \alpha_j \;\; \forall \;\; (\{x_i\}, y) \in \mathcal{D} \right\} \right| \tag{521}$$

For the case of binary classification the "+1" label is said to be the "positive one, and "-1" the "negative". Therefore, by ordering the values so that $\alpha_1 = +1$ and $\alpha_2 = -1$, the confusion matrix would present the occurrence frequency of true positives (TP), false negatives (FN), false positives (FP) and true negatives in the (1,1), (1,2), (2,1) and (2,2) entries, respectively.

y \ $\hat{y}$	+1	-1
+1	TP	FN
-1	FP	TN

Note that in the general case the sum of the entries of the i-th row gives the actual occurrence frequency of the i-th class. And the sum of the entries of the j-th column gives the predicted occurrence frequency of the j-th class.

Precision and Recall

Precision associated with the i-th class corresponds to the ratio between the number of correct predictions of the i-th class and the total number of predictions of that class:

$$\text{precision}_i = \frac{f_{ii}}{\sum_{j=1}^{K} f_{ji}} \tag{522}$$

And recall associated with the i-th class corresponds to the ratio between the number of number of correct predictions of the i-th class and the number of actual occurrences of that class:

$$\boxed{\text{recall}_i = \frac{f_{ii}}{\sum_{j=1}^{k} f_{ij}}} \tag{523}$$

From (522) one concludes that precision is maximum (1) if all predictions of the i-th class are correct. However, that does not mean that the model predicts all occurrences of i-th class. If that was true, the number of predictions of the i-th class $\sum_{j=1}^{k} f_{ji}$ would also need to be equal to the number of actual occurrences of that class, $\sum_{j=1}^{k} f_{ij}$, and the model would be said to be accurate with respect to the i-th class besides precise. Precision is also termed "positive predictive value"

From (523) one concludes that recall is maximum (1) if all actual occurrences of the i-th class are predicted by the model. The term "recall" comes from the fact that a model that satisfies such condition is able to "retrieve" all occurrences of the class. Note that though a model might have recall 1 it is not necessarily accurate since it might predict the i-th class when the j-th one actually occurred (the case of false positives). Recall is also termed "sensitivity", "hit rate" and

"true positive rate".

Probabilistically one may define precision and recall as

$$\text{precision}_i = P(y = \alpha_i \mid \hat{y} = \alpha_i) \overset{(52)}{=} \frac{P(y = \alpha_i, \hat{y} = \alpha_i)}{P(\hat{y} = \alpha_i)}$$

$$\overset{(49)}{=} \frac{P(y = \alpha_i, \hat{y} = \alpha_i)}{\sum_{j=1}^{k} P(y = \alpha_j, \hat{y} = \alpha_i)}$$

$$\text{recall}_i = P(\hat{y} = \alpha_i \mid y = \alpha_i) \overset{(52)}{=} \frac{P(y = \alpha_i, \hat{y} = \alpha_i)}{P(y = \alpha_i)} =$$

$$\overset{(49)}{=} \frac{P(y = \alpha_i, \hat{y} = \alpha_i)}{\sum_{j=1}^{k} P(y = \alpha_i, \hat{y} = \alpha_j)}$$

For the case of binary classification precision and recall are usually expressed with respect to the positive class "1" and given by

$$\boxed{\text{precision} := \text{precision}_+ = \frac{TP}{TP + FP}} \tag{524}$$

$$\boxed{\text{recall} := \text{recall}_+ = \frac{TP}{TP + FN}} \tag{525}$$

Usually it is not possible to have both high and precision. There is a tradeoff between the two. Note that the refinement of the scores is widely associated with false positives (in the case of precision) and with false negatives (in the case of recall).

- An example in which false positives are to be minimised - and, therefore, precision maximised - is the identification of spam messages. It would be really inconvenient to have an important message to be falsely labelled as spam making the user to not read it

- An example in which false negatives are to be minimised - and, therefore, recall maximised - is the diagnosis of cancer. Falsely diagnosing a patient to not have cancer when in truth he/she has might lead to his/her death.

Accuracy

Accuracy of the model is defined as the ratio between the number of correct predictions and total number of predictions, or equivalently the ratio between the trace of the confusion matrix and the sum of all elements of the matrix:

$$\text{accuracy} = \frac{\sum_{i=1}^{K} f_{ii}}{\sum_{i=1}^{K}\sum_{j=1}^{K} f_{ij}} \tag{526}$$

Probabilistically, on may define accuracy as

$$\text{accuracy} = P(\hat{y}=y) = \sum_{i=1}^{K} P(\hat{y}=y, y=x_i) =$$

(99)

$$= \sum_{i=1}^{K} P(y=x_i, \hat{y}=x_i)$$

In the case of binary classification (526) is equivalent to

$$\text{accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{527}$$

Area under the ROC (AUC)

Let one consider binary classification. Let TPR be the true positive rate, that is, the ratio between true positives and actual positives (which is equivalent to recall):

$$TPR = \frac{TP}{TP+FN} = \text{recall} \tag{528}$$

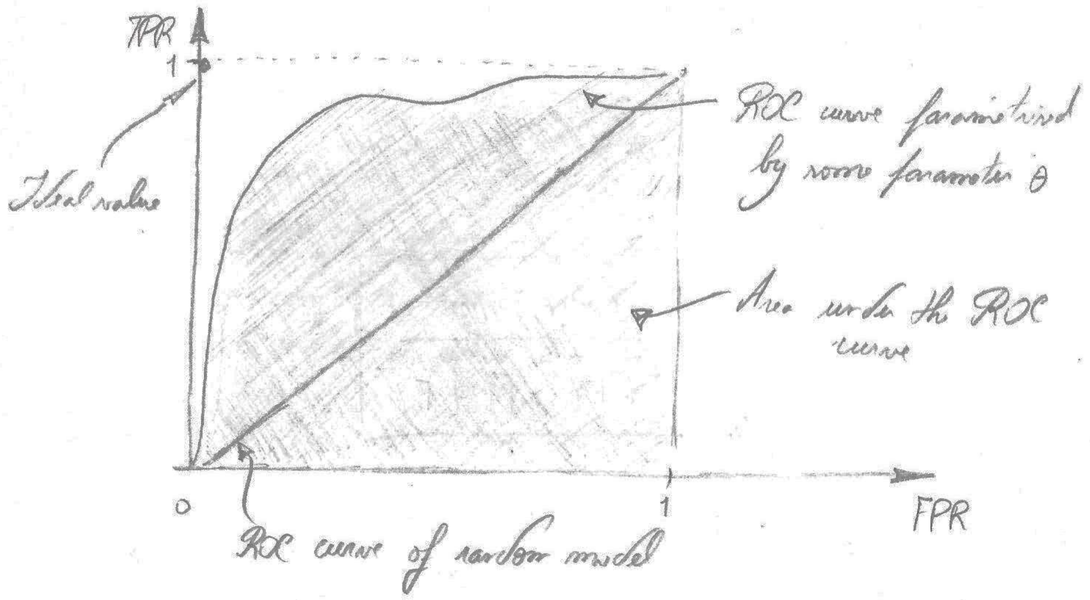
(True positive rate)

And let FPR be the false positive rate, that is, the ratio between false positives and actual negatives:

$$FPR = \frac{FP}{FP+TN} \tag{529}$$

(False negative rate)

The ROC curve (standing for "Receiver Operating Characteristic" - historically used in radar engineering) is defined as the function of TPR on FPR when varying some parameter θ of the model.



The ideal model would be one that does not return any false positive or false negative. With FP=0 and FN=0 one gets from (529) and (528) that FPR=0 and TPR=1 (the point at the upper left corner).

It may be shown that the line TPR = FPR is associated with a random classifier:

Let m_+ be the number of actual positives, $m_- = m - m_+$ the number of actual negatives. And let γ be the probability of the random classifier classifying a case as positive.

A fraction γ of actual positives would then be correctly labelled. And a fraction γ of actual negatives would be incorrectly labelled. Therefor,

$$TP = \gamma m_+$$
$$FN = (1-\gamma) m_+$$
$$TN = (1-\gamma)(m - m_+)$$
$$FP = \gamma (m - m_+)$$

$$TPR = \frac{TP}{TP+FN} = \frac{\gamma m_+}{\gamma m_+ + (1-\gamma) m_+} = \gamma \quad (528)$$

$$FPR = \frac{FP}{FP+TN} = \frac{\gamma (m - m_+)}{\gamma (m - m_+) + (1-\gamma)(m - m_+)} = \gamma = TPR \quad (529)$$ □

From this fact it may be understood that models with a ROC below the line TPR = FPR are not appropriate. Indeed for TPR < FPR:

$$TPR < FPR \Leftrightarrow \frac{TP}{TP+FN} < \frac{FP}{FP+TN} \quad (a)$$

(528) (529)

$$\Leftrightarrow TP \cdot FP + TP \cdot TN < TP \cdot FP + FN \cdot FP \Leftrightarrow$$

$$\Leftrightarrow \boxed{FP \cdot FN > TP \cdot TN} \quad (530)$$

Result (530) shows that the product of the numbers of the two kind of errors is higher than the product of the numbers of the two kind of correct predictions

in the case of the model being below the line TPR = FPR.

The ideal model would have the highest Area Under the ROC Curve (AUC) being this 1.

In the case of the random classifier, AUC = 0.5.

A proper model should have

$$AUC > 0.5.$$

Regression performance metrics

The value of the objective function whose estimation is to be sought may be used to measure the quality of the regression model. One common objective function is the mean squared error (MSE), that is, the arithmetic average of the squares of the errors:

$$MSE = \overline{(\hat{y}_i - y_i)^2} = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 \quad (531)$$

(Mean squared error)

Another measure of the quality of the regression model is the coefficient of determination:

$$R^2 = 1 - \frac{RSS}{TSS}, \quad (532)$$

(Coefficient of determination)

where

$$RSS = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 = m\,MSE \quad (533)$$

is the residual sum of squares and

$$TSS = \sum_{i=1}^{m} (y_i - \bar{y})^2 = m\,s^2 \quad (534)$$

is the total sum of squares. The residual sum of squares is the mean squared error scaled by the number of sample points, m. And the total sum of squares is the uncorrected sample variance (see equation (96) for the case of the covariance matrix) scaled by m. This uncorrected sample variance is given by

$$s^2 = \frac{1}{m} \sum_{i=1}^{m} (y_i - \bar{y})^2. \quad (535)$$

A model with null MSE would be associated with $R^2 = 1$. A model where MSE is the same as s^2 (for instance, if the model is the arithmetic mean $\hat{y}_i = \bar{y}$), it would have $R^2 = 0$. If $MSE < s^2$, then, $R < 0$.

XIII - Neural Networks

Introduction to "vanilla" neural networks

The "vanilla" neural networks are the so-called feed-forward neural networks (FNN) in which information travels unidirectionally - from inputs to outputs through the units in between without going into any loop.

The neural network is comprised by units which perform particular operations on the information that flows into them. The units are organised by layers. In vanilla neural networks each unit of a layer is connected to each unit of the following one - the layers are then said to be fully-connected. The units of the input layer are represented by circles and the information that flows into them are the components of the feature vector $\{x_j\}$. The j-th unit of the input layer then receives the j-th component of the feature vector, $x^{(j)}$. The input units do not transform the information - the respective operation is then the identity. The information then proceeds to the following layers. Let us assume that this is not the output one. The u-th unit of that layer offers a linear transformation to the input vector $\{\tilde{x}_j\}: \{x_j\} \to z_2^{(u)} = \{w_2^{(u)}\}^T \{x_j\}$ (here convention the convention that $\{x_j\}$ is $\{x_j\}$ with an extra a pseudo-component associated with the value 1). It then transforms $z_2^{(u)}$ into the value of some activation function g_2, outputting $a_2^{(u)} = g_2(z_2^{(u)})$...

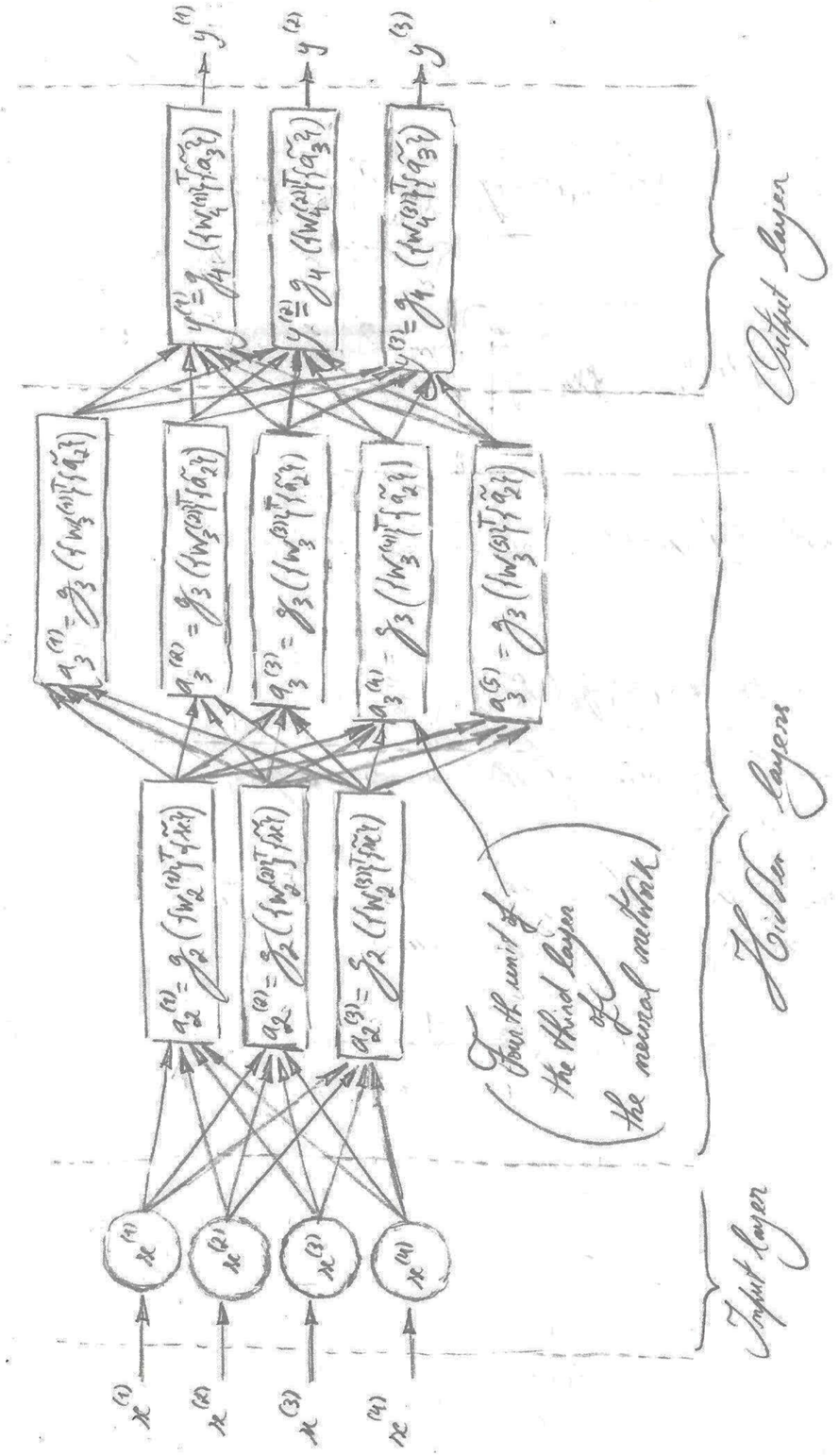
The combination of the outputs of this layer, $\{a_2\} = \{a_2^{(1)} \dots a_2^{(U_2)}\}$ where U_2 is the number of its units becomes together with the unity (to account a bias) the input of the following layer. Let the combination be $\{a_2^*\}$.

Analogous transformations follows from layer to layer until the output layer is reached. Let L be the number of layers of the neural network. If the problem is regression there would be a single unit in the output layer which would compute the so-wanted label. If the problem is classification then there would be an unit per class - the predicted class would have its unit "more activated" than the others - or, if the classification is binary, the output layer could be solely comprised by a single unit whose activation may be associated with one class and the non-activation with the other. The layers between the input and the output layers are usually termed "hidden layers".

If the number of hidden layers is greater than 2 then the problem is said to be of deep learning.

The units of hidden and output layers are usually represented by rectangles.

In the following page the example of a vanilla neural network is depicted.



The output of the l-th layer corresponds to

$$\{a_l\} = \{g_l(\{z_l\})\} = \{g_l([W_l]^T\{\tilde{a}_{l-1}\})\}, \tag{536}$$

where

$$[W_l] = \left[\{W_l^{(1)}\} \cdots \{W_l^{(U_l)}\}\right], \tag{537}$$

is the matrix of weights associated with the l-th layer, and

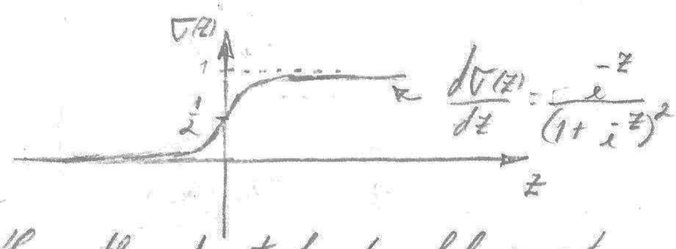
$$\{g_l(\{z_l\})\} = \{g_l(z_l^{(1)}) \cdots g_l(z_l^{(U_l)})\} \tag{538}$$

is the vectorised form of the units' outputs.

Activation functions.

A common activation function that had been used in the past is the sigmoid function (459):

$$g_l(z) = \sigma(z) = \frac{1}{1 + e^{-z}}, \tag{539}$$

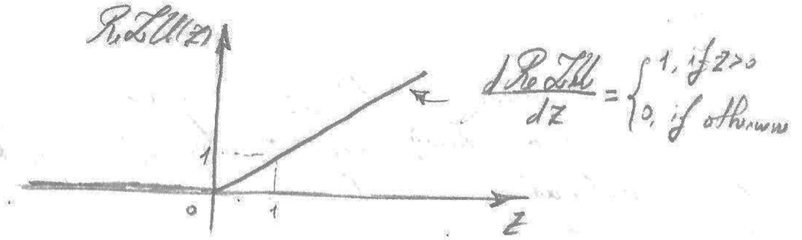


With the advent of powerful computer came the possibility of using neural networks with many layers. However, sigmoid functions are difficult to train in such conditions — they are not sparse (non-null values are always obtained even though they might be small) and their gradients might be quite small — there is the so-called problem

of "vanishing gradients". Indeed, for very large positive or negative values of z the derivative $\frac{d\sigma}{dz}$ is almost null. That is why a different activation function has been used instead of the sigmoid one: the Rectifier Linear Unit (ReLU):

$$g_l(z) = \mathrm{ReLU}(z) = \max(0, z) = \frac{z + |z|}{2} = \begin{cases} z, & \text{if } z > 0 \\ 0, & \text{if otherwise} \end{cases}. \tag{540}$$

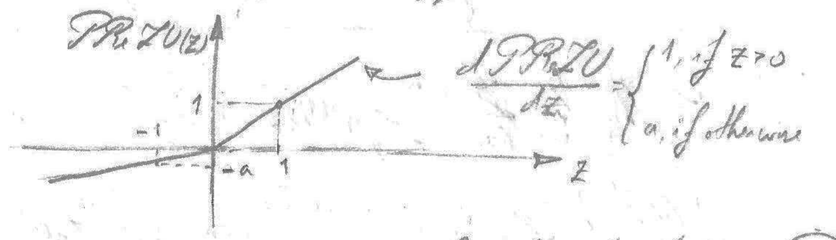
(Rectifier Linear Unit (ReLU))



The ReLU function usually leads to sparse results (with a lot of zeros) and its gradient does not vanish for $z > 0$. It does however when $z < 0$. That is why other variants of the ReLU function have been also used such as the Parametric ReLU:

$$g_l(z) = \mathrm{PReLU}(z) = \begin{cases} z, & \text{if } z > 0 \\ az, & \text{if otherwise} \end{cases}. \tag{541}$$

(Parametric ReLU (PReLU))



where a is a parameter that might even be learnt while training the neural network.

There is a quite famous particularisation of the Parametric ReLU function termed "Leaky ReLU" which sets $a=0.01$:

$$g_e(z) = \text{Leaky ReLU}(z) = \begin{cases} z & \text{if } z>0 \\ 0.01z & \text{if } z<0 \end{cases} \tag{542}$$

(Leaky ReLU function)

Number of weight components of fully-connected NN

The second layer has the same number of weight components has the number of feature components, D, plus one (to account a bias) scaled by the number of units of the second layer.

$$N_2 = (D+1)U_2 \tag{543}$$

In general, the number of weight components of the l-th layer is the number of units of the $l-1$-th layer plus one (to account a bias) scaled by the number of units of the l-th layer:

$$N_l = (U_{l-1}+1)U_l \tag{544}$$

The total number of weight components of the neural network is then

$$N = \sum_{l=2}^{L} N_l = (D+1)U_2 + \sum_{l=3}^{L} (U_{l-1}+1)U_l \tag{545}$$

(Number of weight components of a fully-connected NN)

Convolutional Neural Networks (CNN)

Convolutional neural networks have been historically used in image processing. An image is a matrix of pixels (bidimensional points). These pixels are in turn described by channels. In the case of black and white images, there is solely one channel which is associated with the grey scale: the brightness intensity. In the case of RGB (red-green-blue) images, there are three channels, each one associated with the brightness intensity of a colour, either red green or blue.

Convolutional neural networks rely on the principle of matrix convolution. In this matrix convolution a matrix of same size as the original one is obtained by applying a filter. This filter is in turn described by a matrix termed filter kernel.

Let a channel of some image be described by the $m \times m$ matrix $[x]$. And let the filter kernel be an $a \times b$ matrix $[w]$. If a and b are odd then the (i,j)-th entry of the convolution between $[w]$ and $[x]$ is defined as

$$z_{ij} = ([w] * [x])_{ij} = \sum_{r=-\frac{a-1}{2}}^{\frac{a-1}{2}} \sum_{s=-\frac{b-1}{2}}^{\frac{b-1}{2}} w_{r+\frac{a+1}{2}, s+\frac{a+1}{2}} \cdot x_{i-r, j-s} \tag{546}$$

((i,j)-th entry of the convolution between the $a \times b$ filter kernel $[w]$ and the matrix $[x]$ assuming a and b to be odd)

If a and b are not odd, then, one may define a convention of summing one more "front term" than "back terms" in both row and column directions:

$$z_{ij} = ([w] * [x])_{ij} = \sum_{r=-\frac{a}{2}+1}^{\frac{a}{2}} \sum_{s=-\frac{b}{2}+1}^{\frac{b}{2}} w_{r+\frac{a}{2},s+\frac{b}{2}} \cdot x_{i-r,j-s} \tag{546}$$

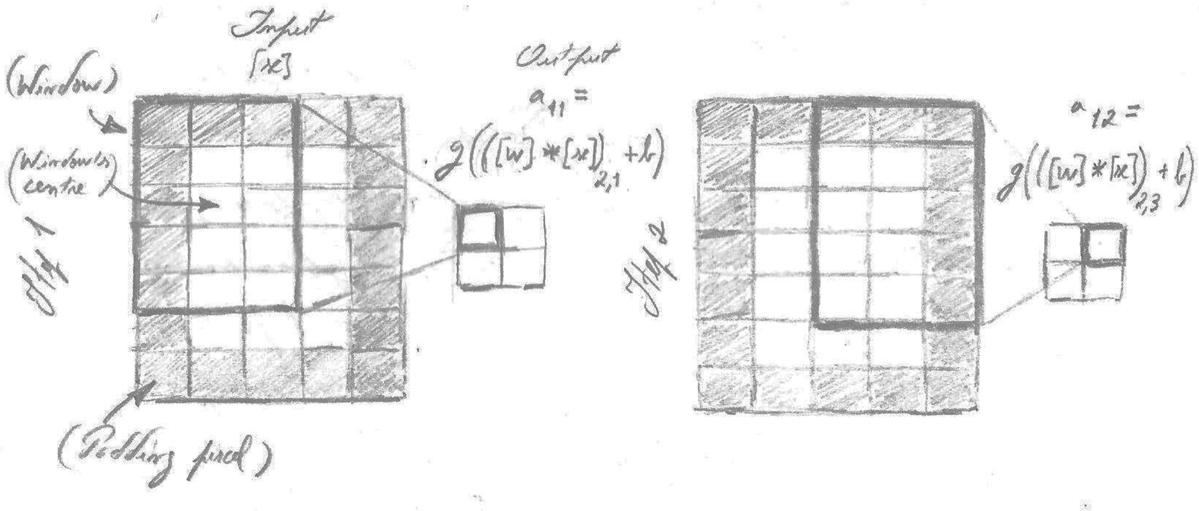
((i,j)-th entry of the convolution between the $a \times b$ filter matrix $[w]$ and matrix $[x]$ assuming a and b to be even)

In the case of a and b being even, the (i,j)-th entry of the convolution defined by (546) corresponds to the Frobenius inner product (the matrix inner product) between the submatrix of size $a \times b$ of $[x]$ centred a (i,j) and the filter kernel $[w]$. The example illustrated below gives the $(4,4)$-th entry of the convolution between a 3×3 kernel matrix and a 5×5 matrix:

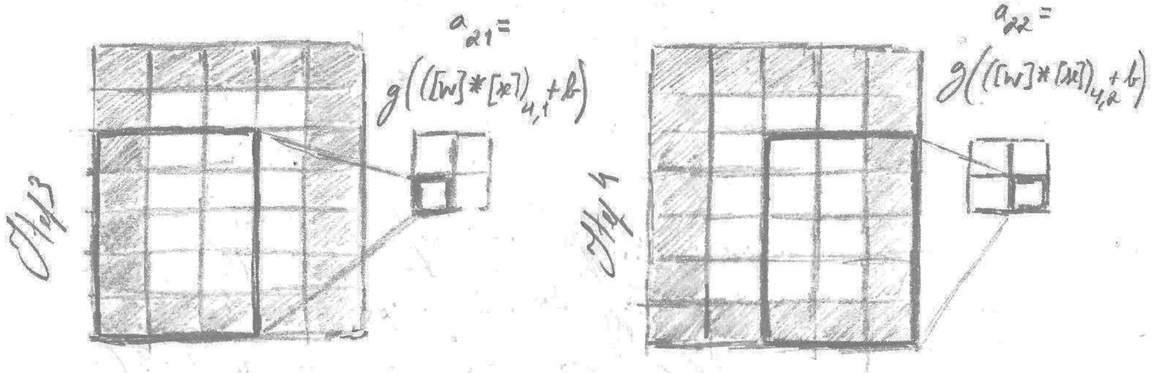
$$\left(\underbrace{\begin{bmatrix} 1 & 2 & 1 \\ 0 & 3 & 0 \\ 2 & 0 & 0 \end{bmatrix}}_{=[w]} * \underbrace{\begin{bmatrix} 1 & 2 & 1 & 3 & 4 \\ 5 & 6 & 7 & 1 & 2 \\ 3 & 3 & 2 & 2 & 3 \\ 4 & 2 & 1 & 2 & 2 \\ 1 & 3 & 0 & 1 & 2 \end{bmatrix}}_{=[x]} \right)_{4,4} = \begin{array}{l} 1\times2 + 2\times2 + 1\times3 + \\ +0\times1 + 3\times2 + 0\times2 + \\ +2\times0 + 0\times1 + 0\times2 \\ = 13 \end{array}$$

In practice, convolution is usually done through the so-called moving window approach in which a "window" representing the filter kernel $[w]$ traverses the matrix $[x]$ from left to right and up to down. The part of the matrix that is bounded by the window is termed "patch" and is the one used for computing a respective entry of the convolution. The window has some stride - the number of pixels it moves in one direction from step to step. Also, padding of the image might be considered. Padding is the number of lines of pixels (usually of channel value 0) that is added to the borders of the image so that the moving window traverses through the borders more times (depending on stride, if there is no padding, the number of times that the window traverses a border pixel is usually lower than the number of times it traverses a core pixel). If the border of the image does not have meaningful information, padding may be disregarded. One important note regarding stride is that when the window finishes traversing a row it goes to the beginning of the next one which departs from the previous by the length of a stride. Another important note is that if stride is greater than 1 and/or the size of the filter kernel is greater than 1×1, there will be entries of the convolution that will be skipped, that is, the size of the matrix resulting from the convolution would be smaller than the size of the image with its padding.

In convolutional neural networks the filter besides performing a convolution, adds a bias term b and applies an activation function g to the result. Below, an example of applying a 5×3 filter of stride 2 and padding 1 to an image of size 5×3 is illustrated.

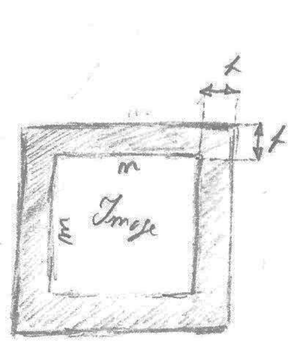


[Note:
In step 1 the window's centre is at coordinate (2,1) of the image. Therefore, the respective coordinate of the convolution is (2,1). Still, due to the size and stride of the filter, the result is at coordinate (1,1) of the output (since it is obtained from the first step).]



One important note when applying some filter is to choose the filter size, stride and padding so that when the moving window traverses the image (and padding) in each direction it goes all the way, that is, without letting some of the pixels near the right and bottom borders to not be traversed. Let the image, filter kernel and output have sizes $m\times m$, $a\times b$ and $u\times v$, respectively. Also, let the stride and padding be s and p, respectively. It may be shown that the output dimensions u and v are given by

$$u = \text{floor}\left(\frac{m+2p-a}{s}+1\right), \tag{548}$$



(Relation between output, image and kernel row size (u, m and a, respectively) as well as stride s and padding p)

$$v = \text{floor}\left(\frac{m+2p-b}{s}+1\right), \tag{549}$$

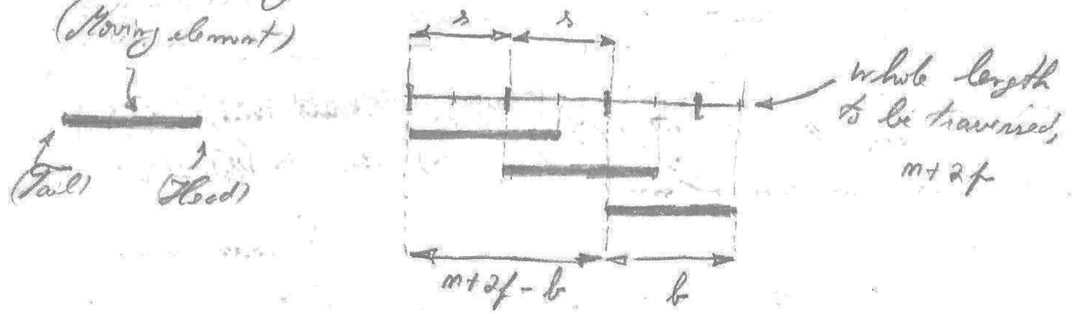


(Relation between output, image and kernel column size (v, m and b, respectively) as well as stride s and padding p)

where "floor" is the floor function. It gives the greatest integer less than or equal to the argument. This is required if the argument is not an integer, then the moving window would not fit into the input in its last step of the direction (row and/or column) and the respective output value would be discarded.

Note: (Proof of (549))

Let one prove (549). Note that a proof of (549) is also a proof of (548) since the motion of the window only differs in direction in the two cases. The proof is actually straightforward. If a moving element is able to traverse all points along a line:



$$p = \frac{n+2b}{s} + 1$$

- $\frac{n+2b}{s}$ = number of divisions of the n-b length in which the tail of the element may be
- $+1$ = length that the tail of the element may traverse
- the last portion of element in the whole length in

In the case of the image having multiple channels - let the number of such channels be c - the filter would have a kernel matrix for each one ($[W_1], \dots, [W_c]$) and a common bias term (b). Let $[x_j]$ stand for the l-th image matrix associated with the j-th channel. The affliction of the filter on the image (the combination of the channels) would be the

sum of the convolutions between the kernels $[W_1], \dots, [W_c]$ and channel values $[x_1], \dots, [x_c]$, respectively, with the bias b added and the activation function g applied to the entries of the result, that is,

$$[a] = [g((\sum_{j=1}^{c} [W_j] * [x_j]) + b[1])] \quad (550)$$

Note: There is a parallelism between the term $(\sum_{j=1}^{c} [W_j] * [x_j]) + b[1]$ and the linear transformation $\sum_{j=1}^{c} w_j x_j + b$

(Output matrix of filter of kernels $[W_1], \dots, [W_c]$ and bias b applied to image channels $[x_1], \dots [x_c]$)

where $[1]$ is a matrix of ones.

Note that in (550) the term $[W_j] * [x_j]$ does not mean a convolution in its whole but the result that would be obtained by using the moving window approach.

Each filter should be regarded in a convolutional neural network as a unit. Let U_l be the number of units (filters) of the l-th layer. The output of the l-th layer would be a combination of U_l matrices of the form (550). Such combination is termed "volume". And the number of its elements, U_l, is termed "volume's depth". Let $[[a_l]]$ denote the output of the l-th layer (a third-order tensor). The u-th entry of such output would be

$$[a_l^{(u)}] = [g_l((\sum_{j=1}^{U_{l-1}} [W_{lj}^{(u)}] * [a_{l-1}^{(j)}]) + b_l^{(u)}[1])] \quad (551)$$

(Output of the u-th filter of the l-th layer)

The weights $[W_{ij}^{(u)}]$ and biases $b_i^{(u)}$ are to be learned when training the neural network.

Usually the convolutional layer is proceeded by a pooling layer. Pooling is analogous to convolution in the sense that it also uses the moving window approach and reduces a patch to a single value. However, instead of transforming the entries of the patch by using trainable parameters, it applies a fixed operation on the set of entries — there are no weights to be trained. Furthermore, pooling of a volume of some depth returns a volume of same depth (in the case of a convolutional layer the depth of the returned would be the number of filters of that layer). Pooling of a volume of matrices is the volume of the poolings — one for each matrix. The most used pooling operations are average pooling, AvgPool — in which the averages of entries of the patches are the entries of the returned matrix — and maximum pooling, MaxPool — in which the maximum of the entries patches are the entries of the returned matrix. Pooling then allows to reduce the dimensionality of the problem to deal with without increasing the number of training parameters.

Note that pooling has several hyperparameters: the size, stride and padding of the moving window and the kind of operation. Padding in the context of pooling is used to make the moving window to traverse all pixels

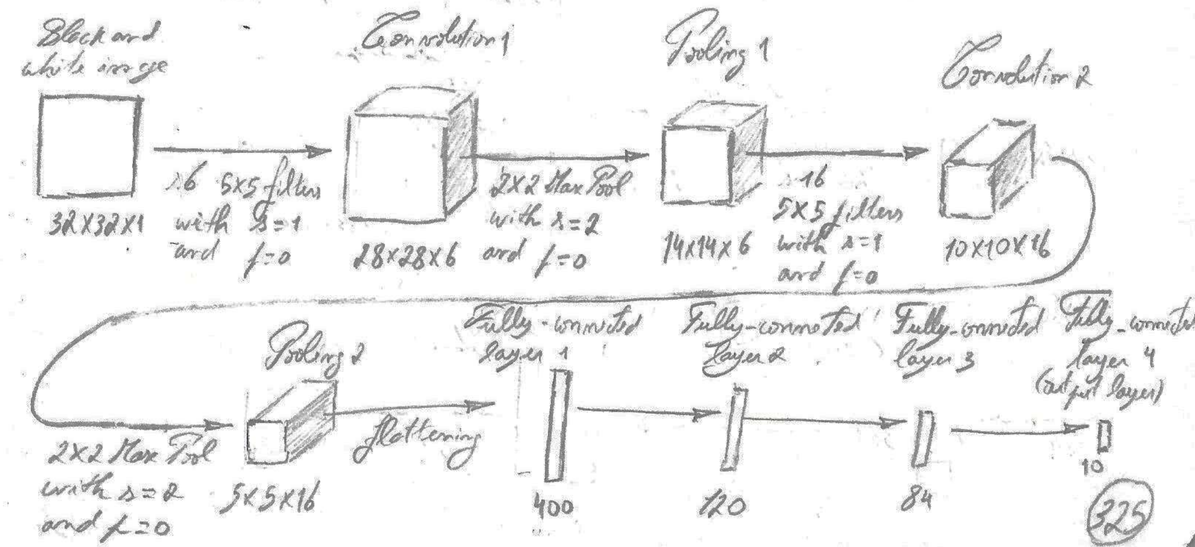
In the figure below the MaxPool operation of size 3×3, stride 3 and null padding is applied to a 6×6 matrix.

$$\text{MaxPool}_{3\times3,\,s=2,\,p=0}\left(\begin{bmatrix} 1 & 2 & 1 & 3 & 4 & 1 \\ 5 & 6 & 7 & 1 & 2 & 2 \\ 3 & 3 & 2 & 2 & 3 & 1 \\ 5 & 2 & 1 & 2 & 2 & 8 \\ 1 & 3 & 0 & 1 & 2 & 7 \\ 0 & 1 & 3 & 4 & 5 & 6 \end{bmatrix}\right) = \begin{bmatrix} 7 & 4 \\ 5 & 8 \end{bmatrix}.$$

The output of the convolutional network — a third-order tensor — is then flattened, that is, transformed into a vector. The vector would become the input layer of a fully connected neural network which would output the to-wanted label.

LeNet-5

An example of a neural network that might be used for classifying handwritten digits is LeNet-5:



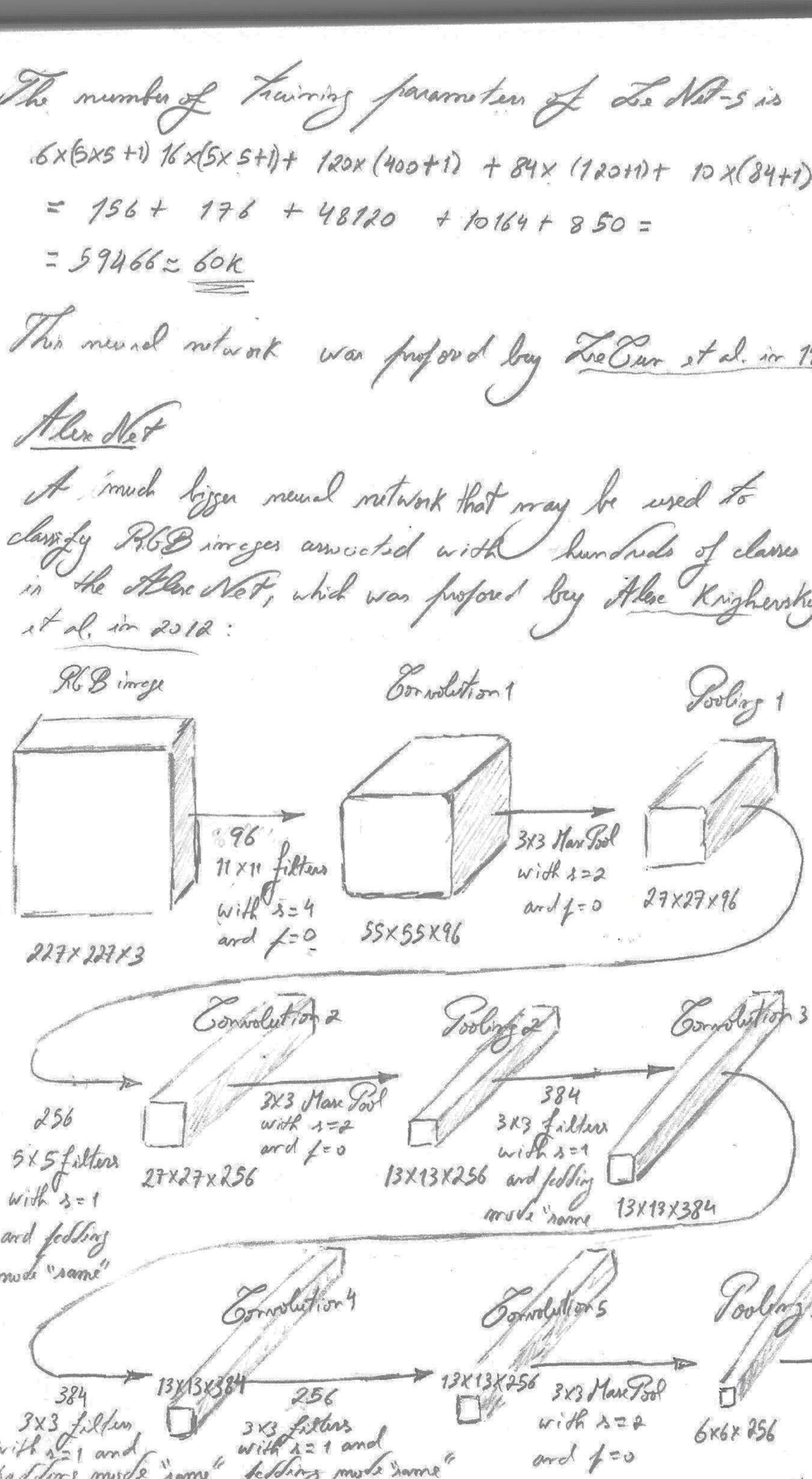
The number of training parameters of LeNet-5 is

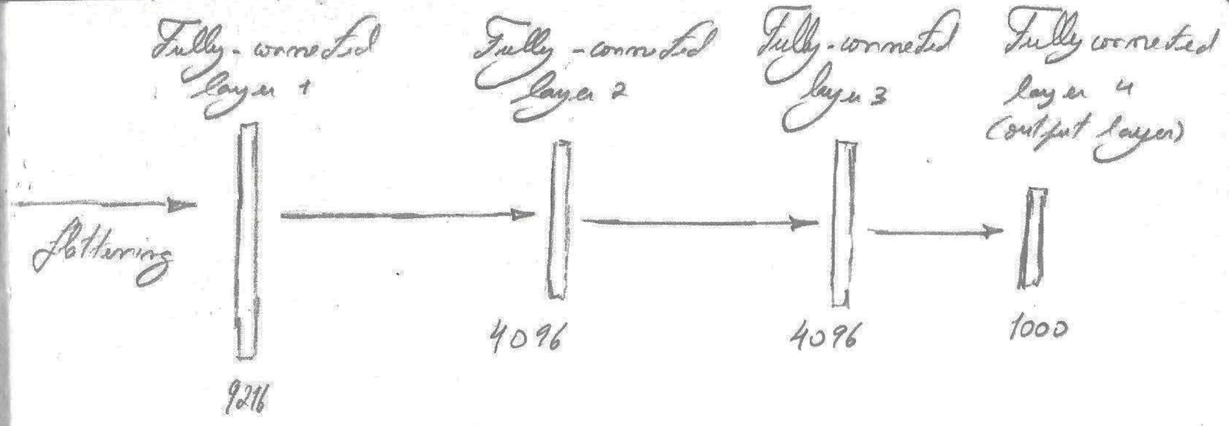
$$6\times(5\times5+1)+16\times(5\times5+1)+120\times(400+1)+84\times(120+1)+10\times(84+1)=$$
$$=156+176+48120+10164+850=$$
$$=59466\simeq 60K$$

This neural network was proposed by LeCun et al. in 1998.

AlexNet

A much bigger neural network that may be used to classify RGB images associated with hundreds of classes is the AlexNet, which was proposed by Alex Krighevsky et al. in 2012:





Note: (Padding mode "same")

In the case of the stride being $s=1$, the padding mode "same" is such that the transformed matrix has the same width and height as the original one. It adds padding layers evenly as possible (in the case of the added layers in one direction being necessarily odd, the last padding layer is usually added to the right margin if the direction is horizontal or to the bottom margin if the direction is vertical). In the case of the stride being $s>1$, the padding mode "same" adds padding layers to the margins as evenly as possible so that all pixels (including the padding ones) are traversed.

The number of training parameters of the AlexNet corresponds to

$$96\times(11\times11+1)+256\times(5\times5+1)+384\times(3\times3+1)+384\times(3\times3+1)+256\times(3\times3+1)+$$
$$+4096\times(9216+1)+4096\times(4096+1)+1000\times(4096+1)=$$
$$=11712+6656+3840+3840+2560+37752832+16781312+$$
$$+4097000=58659752\simeq 60M.$$

XIV - Gradient Descent

"Vanilla" Gradient Descent (or "batch" Gradient Descent)

Given the dataset $\mathcal{D} = \{(\{x_1\}, y_1), \dots (\{x_n\}, y_n)\}$, the Machine Learning problem is to find the extremum point of some objective function. Such objective function may anyway be transformed into a cost function whose minimum is to be found. Let

$$\boxed{f(\{w\}) = \frac{1}{m}\sum_{i=1}^{m} f_i(\{w\})} \qquad (552)$$

be the cost function and $\{w\}$ the respective set of parameters. The term $\frac{1}{m} f_i(\{w\})$ is the loss function associated with the i-th data point. The Machine Learning problem is then

$$\min_{\{w\}} f(\{w\}).$$

A method for finding the minimiser $\{w^*\}$ is Gradient Descent. The "vanilla" form of this method consists on initialising $\{w\}$ and successively adding the gradients of f scaled by some negative constant, $-\gamma$, until the components of $\{w\}$ start varying - convergence is achieved.

The algorithm may then be expressed as

1. Initialize the set of parameters: $\{w\} = \{w_0\}$

2. For $t = 1, \dots$ until $\|\{w_t\} - \{w_{t-1}\}\| < \varepsilon$, where $\varepsilon > 0$ is some arbitrated constant,

   2.1 Compute the gradient of f at $\{w\} = \{w_{t-1}\}$, that is, $\{\nabla f(\{w_{t-1}\})\}$

   2.2 Update the set of parameters:

$$\boxed{\{w_t\} = \{w_{t-1}\} - \gamma \{\nabla f(\{w_{t-1}\})\}} \quad (553)$$

   where $\gamma > 0$ is some arbitrated constant, called learning rate.

3. Output $\{w^*\} = \{w_t\}$.

The effect of the update (553) is quite intuitive. The motivation behind (553) is to find a $\{w\}$ value close enough to $\{w\} = \{w_{t-1}\}$ so that the first order Taylor series of f around $\{w\} = \{w_{t-1}\}$ may be used as an approximation of f and simultaneously satisfy $f(\{w_t\}) \leq f(\{w_{t-1}\})$. Indeed

$$f(\{w\}) = f(\{w_{t-1}\}) + \{\nabla f(\{w_{t-1}\})\}^T (\{w\} - \{w_{t-1}\}) + O(\|\{w\} - \{w_{t-1}\}\|^2)$$

$$\approx f(\{w_{t-1}\}) + \{\nabla f(\{w_{t-1}\})\}^T (\{w\} - \{w_{t-1}\})$$

Therefore,

$$f(\{w_t\}) \leq f(\{w_{t-1}\}) \Leftrightarrow \boxed{\{\nabla f(\{w_{t-1}\})\}^T (\{w_t\} - \{w_{t-1}\}) \leq 0} \quad (554)$$

From (554) one understands that if $\{\nabla f(\{w_{t-1}\})\} \neq \{0\}$, $\{w_t\}$ would need to be such that $\{w_t\} - \{w_{t-1}\}$ and $\{\nabla f(\{w_{t-1}\})\}$ make an angle higher than or equal to $\frac{\pi}{2}$. Furthermore, the most close to π the angle between the vectors gets, the higher the difference between $f(\{w_t\})$ and $f(\{w_{t-1}\})$. Therefore, in "vanilla" Gradient Descent, $\{w_t\}$ is defined so that $\{w_t\} - \{w_{t-1}\}$ makes an angle of π with $\{\nabla f(\{w_{t-1}\})\}$ while maintaining the first order approximation valid:

$$\{w_t\} - \{w_{t-1}\} = -\gamma \{\nabla f(\{w_{t-1}\})\} \Leftrightarrow \{w_t\} = \{w_{t-1}\} - \gamma \{\nabla f(\{w_{t-1}\})\}$$

($\Leftarrow$ $\{w\} = \{w_t\}$)

with $\gamma > 0$. To ensure that the first order approximation remains valid the coefficient should be quite small.

Convergence is achieved when $f(\{w_t\}) - f(\{w_{t-1}\}) \to 0 \Leftrightarrow$

$\Leftarrow$ $\|\{w_t\} - \{w_{t-1}\}\| \to 0$.

($f(\{w_t\}) \approx f(\{w_{t-1}\}) + \{\nabla f(\{w_{t-1}\})\}^T (\{w_t\} - \{w_{t-1}\})$
$= f(\{w_{t-1}\}) - \frac{1}{\gamma} \|\{w_t\} - \{w_{t-1}\}\|^2$
(553))

An iteration of the "vanilla" Gradient Descent algorithm is said to be an "epoch". The term "epoch" denotes one usage of the whole dataset in the optimisation approach. Since the whole dataset is used to compute the gradient in each iteration (553) (f is given by the whole series in (552)) each iteration is

an "epoch".

Backtracking Gradient Descent

One inconvenience of the "vanilla" Gradient Descent algorithm is the choice of a proper learning rate γ. It would be better for the algorithm to be adaptative and choose γ so that it makes big steps in $\{w\}$ when the induced decrease is slower than the expected and smaller ones when it is faster. A variant of gradient descent that does that is one with "backtracking line search". The respective algorithm is

1. Initialize the set of parameters and the learning rate:

$$\{w\} \leftarrow \{w_0\} \qquad \gamma = \gamma_0 > 0$$

2. For $t = 1, \dots$ until $\|\{w_t\} - \{w_{t-1}\}\| < \varepsilon$ where $\varepsilon > 0$ is some arbitrated constant.

2.1. Compute the gradient of f at $\{w\} = \{w_{t-1}\}$, that is, $\{\nabla f(\{w_{t-1}\})\}$.

2.2. If

$$\boxed{\left\| f\left(\{w_{t-1}\} - \gamma_{t-1}\{\nabla f(\{w_{t-1}\})\}\right) \leq f(\{w_{t-1}\}) - c\gamma_{t-1}\|\{\nabla f(\{w_{t-1}\})\}\|^2 \right\|} \quad (555)$$

(Armijo - Goldstein condition)

where $c \in]0,1[$ is some arbitrated constant, then set

$$\boxed{\gamma_t = \tau \cdot \gamma_{t-1}}, \quad (556)$$

where $\tau \in]0,1[$ is some arbitrated constant.

Else, set

$$\boxed{\gamma_t = \gamma_{t-1}} \quad (557)$$

2.3 Set

$$\boxed{\{w_t\} = \{w_{t-1}\} + \gamma_t\{\nabla f(\{w_{t-1}\})\}} \quad (558)$$

3. Output $\{w^*\} = \{w_t\}$.

Usual values of c and τ are $c = \frac{1}{2}$ and $\tau = \frac{1}{2}$.

Note that from condition (559) and the actions (556) and (557), one concludes that γ may either decrease or not change value in an iteration. Therefore, the initial value for γ, γ_0, should not be too small.

Let one analyse the Armijo-Goldstein condition (555).

The term on the left-hand side would be the new value for the function f if the old learning rate γ_{t-1} was kept. The term on the right-hand side is the previous value for f subtracted by some threshold that is proportional to the old learning rate γ_{t-1} and the square of the norm of the gradient evaluated at $\{w\} = \{w_{t-1}\}$, that is, $\|\{\nabla f(\{w_{t-1}\})\}\|^2$. If the decrease that γ_{t-1} would induce is big enough, then, a smaller learning rate given by $\gamma_t = \tau \cdot \gamma_{t-1}$ would be used instead. If the decrease is not big enough, then the old learning rate γ_{t-1} would be kept.

Furthermore,

$$\boxed{f\left(\{w_{t-1}\} - \gamma_{t-1}\{\nabla f(\{w_{t-1}\})\}\right) = f(\{w_{t-1}\}) - \gamma_{t-1}\|\{\nabla f(\{w_{t-1}\})\}\|^2 + \mathcal{O}\left(\|\{\nabla f(\{w_{t-1}\})\}\|^2\right)} \tag{559}$$

Therefore, (555) is equivalent to

$$-\gamma_{t-1}\|\{\nabla f(\{w_{t-1}\})\}\|^2 + \mathcal{O}(\|\{\nabla f(\{w_{t-1}\})\}\|^2) \leq -c\gamma_{t-1}\|\{\nabla f(\{w_{t-1}\})\}\|^2$$

Since $c \in]0,1[$, one sufficient condition for satisfying (555) is to have $\mathcal{O}(\|\{\nabla f(\{w_{t-1}\})\}\|^2) \leq 0$.

"Vanilla" Stochastic Gradient Descent (SGD)

If the number of data points is too big, then, the computation of the gradient $\{\nabla f\}$ may be too costly. In the "vanilla" Stochastic Gradient Descent one approximates the gradient of the cost function, $\{\nabla f\} = \frac{1}{n}\sum_{i=1}^{n}\{\nabla f_i\}$ by the gradient of the loss function associated with a randomly picked single data point, $\frac{1}{n}\{\nabla f_i\}$.

The "stochastic" term comes from the presence of such randomness.

The algorithm is described below.

1. Initialize the set of parameters: $\{w\} = \{w_0\}$
2. For $t = 1, \ldots$ until $\|\{w_t\} - \{w_{t-1}\}\| < \varepsilon$, where ε is some arbitrated constant:
   - 2.1. Randomly shuffle the points $(\{x_i\}, y_i)$ of the dataset $\mathcal{D}$
   - 2.2. Define $\{w_{t-1,0}\} = \{w_{t-1}\}$
   - 2.3 For $i = 1, \ldots, n$:
     - 2.3.1. Compute the n-scaled gradient of the loss function associated with the i-th point at $\{w\} = \{w_{t-1,i-1}\}$, that is, $\{\nabla f_i(\{w_{t-1,i-1}\})\}$.
     - 2.3.2. Update the set of parameters:

$$\boxed{\{w_{t-1,i}\} = \{w_{t-1,i-1}\} - \gamma\{\nabla f_i(\{w_{t-1,i-1}\})\}} \tag{560}$$

where $\gamma > 0$ is the arbitrated learning rate

   - 2.4. Update the set of parameters: $\{w_t\} = \{w_{t-1,n}\}$.

Each iteration in t is termed "epoch" since it uses the whole dataset to go through in subiterations.

Let one consider the example of SGD applied to Linear Regression with $\{x\}$ being unidimensional. From (389), one understands that the loss function associated with the i-th data point is a convex parabola:

$$\boxed{\frac{1}{n}f_i(w) = (w x_i - y_i)^2} \tag{561}$$

This function has its minimum, 0, at

$$\boxed{w_i^* = \frac{y_i}{x_i}} \tag{562}$$

The cost function $f = \frac{1}{m}\sum_{i=1}^{m} f_i$ is also convex and has its minimum at

$$(395):\quad w^* = ([A][A]^T)^{-1}[A]\{b\} = \left([x_1 \dots x_m]\begin{Bmatrix} x_1 \\ \vdots \\ x_m \end{Bmatrix}\right)^{-1}[x_1 \dots x_m]\begin{Bmatrix} y_1 \\ \vdots \\ y_m \end{Bmatrix} =$$

$$= \left(\sum_{i=1}^{m} x_i^2\right)^{-1}\left(\sum_{i=1}^{m} x_i y_i\right) \Rightarrow \boxed{w^* = \frac{\sum_{i=1}^{m} x_i y_i}{\sum_{i=1}^{m} x_i^2}} \tag{563}$$

Let $\boxed{\max(\{w_1^*, \dots, w_m^*\}) = \frac{B}{A}}$ and $\min\boxed{(\{w_1^*, \dots, w_m^*\}) = \frac{b}{a}}$. (564) (565)

One may show that $\frac{b}{a} \le w^* \le \frac{B}{A}$, that is, the optimum weight of the cost function is bounded by the minimum and maximum weight of the loss function.

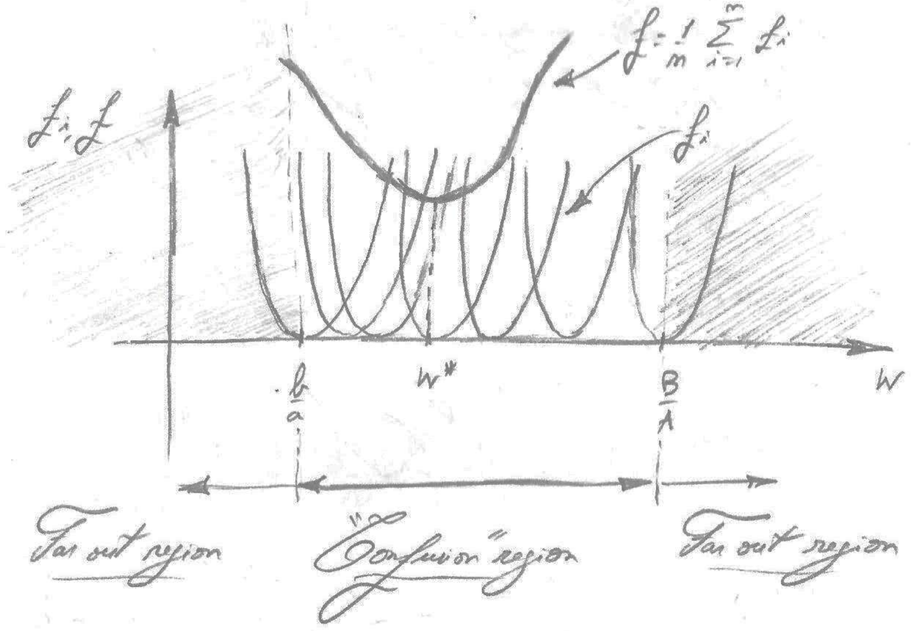
Indeed,

$$\frac{b}{a} \le w_i^* \le \frac{B}{A} \overset{(562)}{\Rightarrow} \frac{b}{a} \le \frac{y_i}{x_i} \le \frac{B}{A} \overset{(x_i^2 \text{ is non-negative})}{\Rightarrow} \frac{b}{a}x_i^2 \le x_i y_i \le \frac{B}{A}x_i^2 \Rightarrow$$

$$\Rightarrow \frac{b}{a}\left(\sum_{i=1}^{m} x_i^2\right) \le \sum_{i=1}^{m} x_i y_i \le \frac{B}{A}\left(\sum_{i=1}^{m} x_i^2\right) \Rightarrow$$

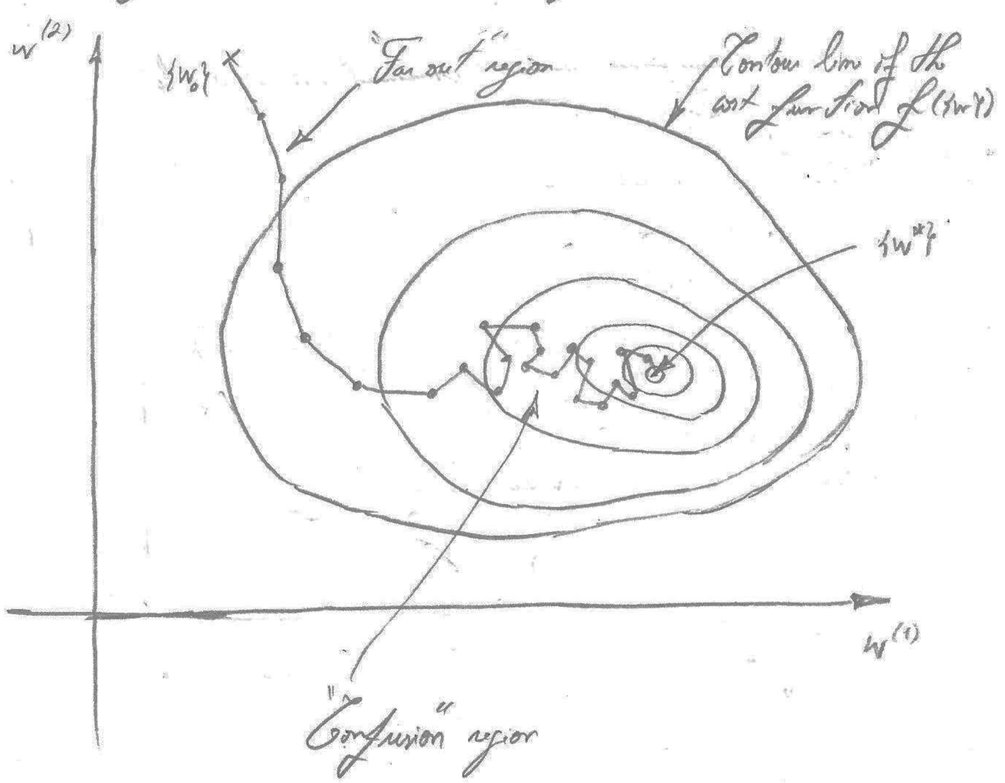
$$\Rightarrow \frac{b}{a} \le \frac{\sum_{i=1}^{m} x_i y_i}{\sum_{i=1}^{m} x_i^2} \le \frac{B}{A} \Rightarrow \boxed{\frac{b}{a} \le w^* \le \frac{B}{A}} \tag{566}$$

$\underbrace{\qquad\qquad}_{= w^*}$



At $w < \frac{b}{a}$ and $w > \frac{B}{A}$ the derivative of the loss functions has necessarily the same sign as the derivative of the cost function as can be concluded from the graph above, that is, $\operatorname{sgn}\left(\frac{d}{dw}\left(\frac{1}{m}f_i\right)\right) = \operatorname{sgn}\left(\frac{d}{dw}f\right)$. In the case of higher dimensions, one could show that the analogous statement would be to say that there is a hypersphere centred at $\{w\} = \{w^*\}$ of some large enough radius for which in the region out of it (the far out region) the gradients of the loss functions and cost function make an angle not greater than $\frac{\pi}{2}$, that is, $\{\nabla \frac{1}{m} f_i\}^T \{\nabla f\} \le 0$. Therefore, the mean of the gradients of the loss functions

may be used in place of the gradient of the cost function to minimise the cost function through Gradient Descent in the far out region. However, in the case of a region close to the optimum weight vector $\{w^*\}$ the angles of the two kinds of gradients are not necessarily smaller or equal to $\frac{\pi}{2}$. Using the gradients of the loss functions as approximation of the gradient of the cost function in such case would make the descent to be foggy and the optimum value may in fact not be achieved - hence the name "confusion region". An example of Gradient Descent using TGD is depicted in the figure below.



"Mini-batch" Stochastic Gradient Descent (Mini-batch SGD)

To improve the accuracy of the approximation for the gradient of the cost function in SGD one may use a sum of the gradients of loss function associated with a batch of data points instead of solely a single data point. This approach is hence termed "Mini-batch SGD" and is described by the algorithm below.

1. Initialise the set of parameters: $\{w\} = \{w_0\}$
2. For $t = 1, \ldots,$ until $\|\{w_t\} - \{w_{t-1}\}\| < \varepsilon$, where ε is some arbitrated constant:
   - 2.1. Randomly shuffle the points (x_i, y_i) of the dataset $\mathcal{D}$
   - 2.2. From the shuffled dataset construct m mini-batches, $\mathcal{D}_1, \ldots, \mathcal{D}_m$.
   - 2.3. Define $\{w_{t-1,0}\} = \{w_{t-1}\}$.
   - 2.4. For $i = 1, \ldots, m$:
     - 2.4.1. Compute the n-scaled sum of the loss functions associated with the batch $\mathcal{D}_i$ at $\{w\} = \{w_{t-1,i-1}\}$, that is, $\sum_{j=1}^{m_i} \{\nabla f_j(\{w_{t-1,i-1}\})\}$ where $m_i = |\mathcal{D}_i|$.
     - 2.4.2. Update the set of parameters:

$$\{w_{t-1,i}\} = \{w_{t-1,i-1}\} - \gamma \sum_{j=1}^{m_i} \{\nabla f_j(\{w_{t-1,i-1}\})\} \quad (567)$$

     where $\gamma > 0$ is the arbitrated learning rate.
   - 2.5. Update the set of parameters: $\{w_t\} = \{w_{t-1,m}\}$

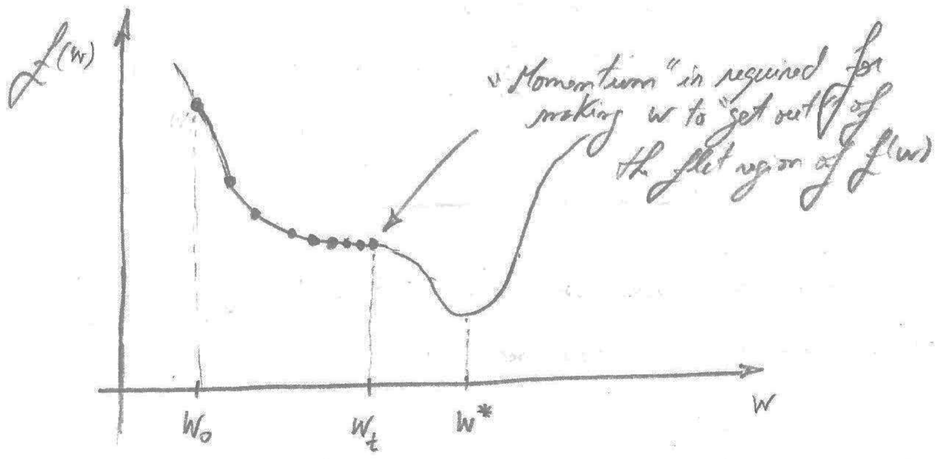
"Momentum" Gradient Descent

The update rule in the "Vanilla" Gradient Descent as presented in (553) might not be able to remove $\{w\}$ from "flat regions" of the cost function since the gradient there is near zero. One way of solving this issue is to add a term to the update rule that accounts how much the parameter had "moved" from its previous place, $\alpha(\{w_{t-1}\}-\{w_{t-2}\})$, so that a new update takes it into consideration. The name "momentum" comes from an analogy of Physics, in which a ball descends a valley, encounters a flat plate, but is able to go through due to the momentum attained during the descent.

Expression (553) is then transformed into

$$\boxed{\{w_t\} = \{w_{t-1}\} - \gamma \{\nabla f(\{w_{t-1}\})\} + \alpha(\{w_{t-1}\}-\{w_{t-2}\})}, \tag{568}$$

where $\alpha \in [0,1]$ is an arbitrated constant.



"Ada Grad" Gradient Descent

AdaGrad is an adaptative Gradient Descent algorithm hence the term "Ada". The AdaGrad adapts the learning rate γ. In fact, it separately adapts different learning rates for different components of the set of parameters, $\{w\}$.

The learning rate for the j-th parameter at the t-th iteration corresponds to the initial given value γ_0 divided by the square root of the sum of the squares of the derivatives of the cost function with respect to the parameter obtained in the previous iterations. Additionally a small factor ε may be added to the denominator to avoid divisions by zero.

Expression (553) is transformed into

$$\boxed{w_t^{(j)} = w_{t-1}^{(j)} - \gamma_t^{(j)} \frac{\partial f}{\partial w^{(j)}}(\{w_{t-1}\}) \text{ for all } j} \tag{569}$$

where

$$\boxed{\gamma_t^{(j)} = \frac{\gamma_0}{\sqrt{\sum_{\tau=0}^{t-1}\left(\frac{\partial f}{\partial w^{(j)}}(\{w_\tau\})\right)^2 + \varepsilon}}} \tag{570}$$

[Note: Some scientists put the "ε" inside the root square.]

From (569) and (570), one concludes that bigger the previous derivatives the smaller the learning rate gets. Note also that the learning cannot increase. It either decreases or maintains the same value. The decrease of the learning rate due to high values of the derivatives in the previous iterations is important for the parameter not skipping the minimum when descending.

Backpropagation

Backpropagation is an algorithm for computing the gradients in a feed-forward neural network. Such neural network is a nested function (a composition of functions). Consider the example in page 313 (a fully connected FNN). A cost function associated with it may be defined as

$$\mathcal{L} = \frac{1}{m}\sum_{i=1}^{m} \| \{a_{L,i}\} - \{y_i\} \|^2 \tag{571}$$

where $a_{L,i}^{(j)}$ and $y_i^{(j)}$ are the estimated and actual values of the label associated with the j-th class of the i-th data point.

Let one consider explicit expression of biases so that (536) may be rewritten as

$$\{a_\ell\} = \{g_\ell(\{z_\ell\})\} = \{g_\ell(\{b_\ell\} + [W_\ell]^T \{a_{\ell-1}\})\}. \tag{572}$$

Therefore, (571) becomes

$$\mathcal{L} = \frac{1}{m}\sum_{i=1}^{m} \| \{g_L(\{b_L\} + [W_L]^T \{g_{L-1}(\{b_{L-1}\} + [W_{L-1}] \cdots \{g_1(\{b_1\} + [W_1]\{x_i\})\} \cdots)\}) \} - \{y_i\} \|^2 \tag{573}$$

The derivative of $\mathcal{L}$ with respect to the N-th component of the weight vector associated with the u-th unit of the ℓ-th layer, $W_\ell^{(u,N)}$ corresponds to

$$\frac{\partial \mathcal{L}}{\partial W_\ell^{(u,N)}} = \frac{\partial}{\partial W_\ell^{(u,N)}} \left(\frac{1}{m}\sum_{i=1}^{m}\sum_{u_L=1}^{U_L} \left(a_{L,i}^{(u_L)} - y_i^{(u_L)}\right)^2 \right) =$$

$$= \frac{2}{m}\sum_{i=1}^{m}\sum_{u_L=1}^{U_L} \left(a_{L,i}^{(u_L)} - y_i^{(u_L)}\right) \frac{\partial a_L^{(u_L)}}{\partial W_\ell^{(u,N)}} \left(z_{L,i}^{(u_L)}\right) \overset{(572)}{=}$$

$$= \frac{2}{m}\sum_{i=1}^{m}\sum_{u_L=1}^{U_L} \left(a_{L,i}^{(u_L)} - y_i^{(u_L)}\right) \underbrace{\frac{dg_L}{dz}\left(z_{L,i}^{(u_L)}\right)}_{= g'^{(u_L)}_{L,i}} \cdot \frac{\partial z_L^{(u_L)}}{\partial W_\ell^{(u,N)}} =$$

$$= \frac{2}{m}\sum_{i=1}^{m}\sum_{u_L=1}^{U_L} \left(a_{L,i}^{(u_L)} - y_i^{(u_L)}\right) g'^{(u_L)}_{L,i} \{W_L^{(u_L)}\}^T \frac{\partial \{a_{L-1}\}}{\partial W_\ell^{(u,N)}} \left(\{z_{L-1,i}\}\right) =$$

$$= \frac{2}{m}\sum_{i=1}^{m}\sum_{u_L=1}^{U_L} \left(a_{L,i}^{(u_L)} - y_i^{(u_L)}\right) g'^{(u_L)}_{L,i} \sum_{u_{L-1}=1}^{U_{L-1}} W_L^{(u_L,u_{L-1})} \frac{\partial a_{L-1}^{(u_{L-1})}}{\partial W_\ell^{(u,N)}} \left(z_{L-1,i}^{(u_{L-1})}\right) \overset{(572)}{=}$$

$$= \frac{2}{m}\sum_{i=1}^{m}\sum_{u_L=1}^{U_L} \left(a_{L,i}^{(u_L)} - y_i^{(u_L)}\right) g'^{(u_L)}_{L,i} \sum_{u_{L-1}=1}^{U_{L-1}} W_L^{(u_L,u_{L-1})} g'^{(u_{L-1})}_{L-1,i} \sum_{u_{L-2}=1}^{U_{L-2}} W_{L-1}^{(u_{L-1},u_{L-2})} \frac{\partial a_{L-2}^{(u_{L-2})}}{\partial W_\ell^{(u,N)}} \left(z_{L-2,i}^{(u_{L-2})}\right) =$$

$$= \frac{2}{m}\sum_{i=1}^{m}\sum_{u_L=1}^{U_L} \left(a_{L,i}^{(u_L)} - y_i^{(u_L)}\right) g'^{(u_L)}_{L,i} \sum_{u_{L-1}}^{U_{L-1}} W_L^{(u_L,u_{L-1})} \cdot g'^{(u_{L-1})}_{L-1,i} \sum_{u_{L-2}=1}^{U_{L-2}} W_{L-1}^{(u_{L-1},u_{L-2})} \cdot g'^{(u_{L-2})}_{L-2,i} \times$$

$$\times \cdots \times \sum_{u_{\ell+1}=1}^{U_{\ell+1}} W_{\ell+2}^{(u_{\ell+2},u_{\ell+1})} \cdot g'^{(u_{\ell+1})}_{\ell+1,i} \cdot W_{\ell+1}^{(u_{\ell+1},u)} g'^{(u)}_{\ell,i} \, a_{\ell-1,i}^{(N)} \overset{(=)}{}$$

$$\Rightarrow \frac{\partial \mathcal{L}}{\partial w_l^{(u,v)}} = \frac{2}{m}\left(\sum_{i=1}^{m} a_{l-1,v}^{(v)}\left[\sum_{u_L=1}^{U_L}\left(a_{L,i}^{(u_L)} - y_i^{(u_L)}\right) g_{L,i}^{\prime\,(u_L)}\left(\sum_{u_{L-1}=1}^{U_{L-1}} w_L^{(u_L,u_{L-1})} \cdot g_{L-1,i}^{\prime\,(u_{L-1})} \times \right.\right.\right.$$

$$\left.\left.\left. \times \cdots \times \left(\sum_{u_{l+1}}^{U_{l+1}} w_{l+2}^{(u_{l+2},u_{l+1})} \cdot g_{l+1,i}^{\prime\,(u_{l+1})} \cdot w_{l+1}^{(u_{l+1},u)} \cdot g_{l,i}^{\prime\,(u)}\right)\cdots\right)\right)\right] \tag{574}$$

And

$$\frac{\partial \mathcal{L}}{\partial b_l^{(u)}} = \frac{2}{m}\left(\sum_{i=1}^{m}\left(\sum_{u_L=1}^{U_L}\left(a_{L,i}^{(u_L)} - y_i^{(u_L)}\right) g_{L,i}^{\prime\,(u_L)}\left(\sum_{u_{L-1}=1}^{U_{L-1}} w_L^{(u_L,u_{L-1})} \cdot g_{L-1,i}^{\prime\,(u_{L-1})} \times \right.\right.\right.$$

$$\left.\left.\left. \times \cdots \times \left(\sum_{u_{l+1}}^{U_{l+1}} w_{l+2}^{(u_{l+2},u_{l+1})} \cdot g_{l+1,i}^{\prime\,(u_{l+1})} \cdot w_{l+1}^{(u_{l+1},u)} \cdot g_{l,i}^{\prime\,(u)}\right)\cdots\right)\right)\right] \tag{575}$$

From (574) and (575), one concludes that the derivative of the cost function $\mathcal{L}$ with respect to a parameter of the u-th unit and l-th layer depends on the derivatives of the respective activation function $g_{l,i}^{\prime\,(u)}$ and all the other ones from the following layer to the last one, that is, $g_{l+1,i}^{\prime\,(1)}, \ldots, g_{L,i}^{\prime\,(U_L)}$, as well as all the weights of the following layer that multiply the result from the u-th unit of the l-th layer, that is, $w_{l+1}^{(1,u)}, \ldots, w_{l+1}^{(U_{l+1},u)}$, and all the weights from the layer succeeding the following one to the last one, that is, $w_{l+2}^{(1,1)}, \ldots, w_L^{(U_L,U_{L-1})}$. This makes the computation of the derivatives with respect to the parameters to be more conveniently done from the last to the first layer.

The auxiliar calculations used in the last layer would be used in the last but one layer and so on. This is the principle for *back propagation*. Note that while going from the last layer to the first one the parameters of the current layer are updated on the ones of the following that would still use the old parameter values of the previous layers to compute the derivatives. This would make all computed derivatives to be associated with the same state.

The back propagation algorithm in some state t would be:

1. Compute $\delta_{L,i,t-1}^{(u)}$ for $i = 1, \ldots, m$ and $u = 1, \ldots, U_L$, as follows

$$\delta_{L,i,t-1}^{(u)} = \frac{\partial \mathcal{L}_i}{\partial z_{L,i}^{(u)}}\left(z_{L,i,t-1}^{(u)}\right) = g_{L,i,t-1}^{\prime\,(u)}\left(\frac{\partial f_i}{\partial a_{L,i}^{(u)}}\right)_{t-1} \tag{576}$$

where f_i is such that the cost function is defined by $\mathcal{L} = \frac{1}{m}\sum_{i=1}^{m} f_i$, and the terms $g_{L,i,t-1}^{\prime\,(u)}$ and $\left(\frac{\partial f_i}{\partial a_{L,i}^{(u)}}\right)_{t-1}$ correspond to

$$g_{L,i,t-1}^{\prime\,(u)} = \frac{dg_L}{dz}\left(z_{L,i,t-1}^{(u)}\right), \tag{577}$$

$$\left(\frac{\partial f_i}{\partial a_{L,i}^{(u)}}\right)_{t-1} = \frac{2 f_i}{\partial a_{L,i}^{(u)}}\left(a_{L,i,t-1}^{(u)}\right). \tag{578}$$

→

1.2. Compute the derivatives of f_i with respect to the parameters of the L-th layer through

$$\left(\frac{\partial f_i}{\partial W_L^{(u,v)}}\right)_{t-1} = \left(\frac{\partial f_i}{\partial a_{L,i}^{(u)}}\right)_{t-1} \left(\frac{\partial a_{L,i}^{(u)}}{\partial z_{L,i}^{(u)}}\right)_{t-1} \left(\frac{\partial z_{L,i}^{(u)}}{\partial W_L^{(u,v)}}\right)_{t-1} \overset{(*)}{=} \boxed{\left(\frac{\partial f_i}{\partial W_L^{(u,v)}}\right)_{t-1} = \delta_{L,i,t-1}^{(u)} \cdot a_{L-1,i,t-1}^{(v)}} \tag{579}$$

$$\left(\frac{\partial f_i}{\partial b_L^{(u)}}\right)_{t-1} = \left(\frac{\partial f_i}{\partial a_{L,i}^{(u)}}\right)_{t-1} \left(\frac{\partial a_{L,i}^{(u)}}{\partial z_{L,i}^{(u)}}\right)_{t-1} \left(\frac{\partial z_{L,i}^{(u)}}{\partial b_L^{(u)}}\right)_{t-1} \overset{(*)}{=} \boxed{\left(\frac{\partial f_i}{\partial b_L^{(u)}}\right)_{t-1} = \delta_{L,i,t-1}^{(u)}} \tag{580}$$

1.3. Update the weights and biases of the L-th layer using some Gradient Descent method. For instance, if "vanilla" full-batch gradient Descent is regarded, then $W_{L,t}^{(u,v)} = W_{L,t-1}^{(u,v)} - \gamma \sum_{i=1}^{m} \left(\frac{\partial f_i}{\partial W_L^{(u,v)}}\right)_{t-1}$ and $b_{L,t}^{(u)} = b_{L,t-1}^{(u)} - \gamma \sum_{i=1}^{m} \left(\frac{\partial f_i}{\partial b_L^{(u)}}\right)_{t-1}$.

2. For $l = L-1, \dots, 1$:

2.1. Compute $\delta_{l,i,t-1}^{(u)}$ for $i = 1, \dots, m$ and $u = 1, \dots, U_l$:

$$\delta_{l,i,t-1}^{(u)} = \left(\frac{\partial f_i}{\partial z_{l,i}^{(u)}}\right)_{t-1} = \sum_{v=1}^{U_{l+1}} \left(\frac{\partial f_i}{\partial z_{l+1,i}^{(v)}}\right)_{t-1} \left(\frac{\partial z_{l+1,i}^{(v)}}{\partial z_{l,i}^{(u)}}\right)_{t-1} \overset{(*)}{=}$$

$$= \delta_{l+1,i,t-1}^{(v)} \qquad = W_{l+1,t-1}^{(v,u)} \, g'^{(u)}_{l,i,t-1}$$

$$\overset{(*)}{=} \boxed{\delta_{l,i,t-1}^{(u)} = g'^{(u)}_{l,i,t-1} \left(\sum_{v=1}^{U_{l+1}} W_{l+1,t-1}^{(v,u)} \, \delta_{l+1,i,t-1}^{(v)} \right)} \tag{581}$$

2.2. Compute the derivatives of f_i with respect to the parameters of the l-th layer through

$$\boxed{\left(\frac{\partial f_i}{\partial W_l^{(u,v)}}\right)_{t-1} = \delta_{l,i,t-1}^{(u)} \cdot a_{l-1,i,t-1}^{(v)}}, \tag{582}$$

$$\boxed{\left(\frac{\partial f_i}{\partial b_l^{(u)}}\right)_{t-1} = \delta_{l,i,t-1}^{(u)}} \tag{583}$$

2.3. Update the weights and biases of the l-th layer using some Gradient Descent method.

Expressions (576), (581), (582) and (583) may be written in a matricial form:

$$\boxed{\{\delta_{L,i,t-1}\} = \left(\sum_{u=1}^{U_L} g'^{(u)}_{L,i,t-1} \{I_u\} \right) \odot \{\nabla_{\{a_{L,i}\}} f_i\}_{t-1}} \tag{584}$$

(Hadamard product)

$$\boxed{\{\delta_{l,i,t-1}\} = \left(\sum_{u=1}^{U_l} g'^{(u)}_{l,i,t-1} \{I_u\} \right) \odot \left([W_{l+1,t-1}]^T \{\delta_{l+1,i,t-1}\} \right)} \tag{585}$$

$$\boxed{\sum_{u=1}^{U_l} \{I_u\} \{\nabla_{\{W_l^{(u)}\}} f_i\}_{t-1}^T = \{\delta_{l,i,t-1}\} \{a_{l-1,i,t-1}\}^T} \tag{586}$$

$$\boxed{\{\nabla_{\{b_l\}} f_i\}_{t-1} = \{\delta_{l,i,t-1}\}}$$

(587)

"RMS Prop" Gradient Descent

"RMS Prop" (standing for Root Mean Square Propagation) is comparable to AdaGrad in the sense that high values of the derivatives make the step to be smaller. Still, it has the advantage of "forgetting" old derivative values which avoids having the step being inconveniently small due to the occurrence of very big derivatives in a far past.

The algorithm corresponds to

$$\boxed{w_t^{(j)} = w_{t-1}^{(j)} - \frac{\gamma}{\sqrt{v_t^{(j)}} + \epsilon} \frac{\partial f}{\partial w^{(j)}}(\{w_{t-1}\})}$$

where $v_t^{(j)}$ is an exponential running average of factor β of the square of the derivative with respect to the j-th component:

$$\boxed{v_t^{(j)} = \beta v_{t-1}^{(j)} + (1-\beta)\left(\frac{\partial f}{\partial w^{(j)}}(\{w_{t-1}\})\right)^2}$$

where $\beta \in [0,1[$ and $v_0^{(j)} = 0$. Let one expand the expression above:

$$v_t^{(j)} = \beta\left(\beta v_{t-2}^{(j)} + (1-\beta)\left(\frac{\partial f}{\partial w^{(j)}}(\{w_{t-2}\})\right)^2\right) + (1-\beta)\left(\frac{\partial f}{\partial w^{(j)}}(\{w_{t-1}\})\right)^2 \Leftrightarrow$$

$$\Leftrightarrow \boxed{v_t^{(j)} = (1-\beta)\sum_{\tau=0}^{t-1} \beta^{t-1-\tau}\left(\frac{\partial f}{\partial w^{(j)}}(\{w_\tau\})\right)^2}$$

From this result one may understand how β may act as a "forgetting" factor for the previous squares of the derivatives. With $\beta \in [0,1[$, the term $\beta^{t-1-\tau}$ is smaller for small τ, meaning that the older terms would be multiplied by a smaller coefficient while the newer ones would be multiplied by larger coefficients. The smaller the β factor, the more significant the forgetting effect is.

The usage of symbol "v" is due to fact of $v_{t-1}^{(j)}$ corresponding to an estimate for the so-called "uncentered variance" of the derivative with respect to the j-th component, that is, $\mathbb{E}\left(\frac{\partial f}{\partial w^{(j)}}\right)^2$. This "uncentered variance" is also designated by second moment of $\frac{\partial f}{\partial w^{(j)}}$ around the value of zero.

Note that in the case of AdaGrad $v_t^{(j)}$ corresponds to $\sum_{\tau=0}^{t-1}\left(\frac{\partial f}{\partial w^{(j)}}(\{w_\tau\})\right)^2$.

"Adam" Gradient Descent

The "Adam" Gradient Descent ("Adam" stands for "Adaptive Moment Estimation") uses both concepts of momentum (the previous gradient directions (with some forgetting factor) have some contribution to the next step) and careful anticipation (the higher the previous gradient components (with some forgetting factor) the smaller the learning rate should be).

The algorithm consists of firstly computing an estimate for the first moment around zero of the derivative $\frac{\partial f}{\partial w^{(j)}}$ using an exponential running average of forgetting factor β_1:

$$\boxed{m_t^{(j)} = \beta_1 m_{t-1}^{(j)} + (1-\beta_1)\frac{\partial f}{\partial w^{(j)}}(\{w_{t-1}\})},$$

with $m_0^{(j)}=0$ and $\beta_1 \in [0,1[$ (note how $m_t^{(j)}$ retains some of the "momentum" of the previous derivatives). It then consists on computing an estimate for the uncentered variance of the derivative $\frac{\partial f}{\partial w^{(j)}}$ using an exponential running average of another forgetting factor β_2:

$$\boxed{v_t^{(j)} = \beta_2 v_{t-1}^{(j)} + (1-\beta_2)\left(\frac{\partial f}{\partial w^{(j)}}(\{w_{t-1}\})\right)^2}$$

with $v_0^{(j)}=0$ and $\beta_2 \in [0,1[$. Both estimates are then corrected to make them unbiased. Indeed,

$$\mathbb{E}(m_t^{(j)}) = \mathbb{E}\left((1-\beta_1)\sum_{\tau=0}^{t-1}\beta_1^{t-1-\tau}\cdot\frac{\partial f}{\partial w^{(j)}}(\{w_\tau\})\right) = (1-\beta_1)\sum_{\tau=0}^{t-1}\beta_1^{t-1-\tau}\mathbb{E}\left(\frac{\partial f}{\partial w^{(j)}}(\{w_\tau\})\right) =$$

$$\simeq (1-\beta_1)\beta_1^{t-1}\left(\sum_{\tau=0}^{t-1}\left(\frac{1}{\beta_1}\right)^\tau\right)\mathbb{E}\left(\frac{\partial f}{\partial w^{(j)}}\right) = (1-\beta_1)\beta_1^{t-1}\frac{1-\left(\frac{1}{\beta_1}\right)^t}{1-\frac{1}{\beta_1}}\mathbb{E}\left(\frac{\partial f}{\partial w^{(j)}}\right) =$$

$$= (1-\beta_1)\frac{1-\beta_1^t}{1-\beta_1}\mathbb{E}\left(\frac{\partial f}{\partial w^{(j)}}\right) = (1-\beta_1^t)\mathbb{E}\left(\frac{\partial f}{\partial w^{(j)}}\right).$$

The analogous result is obtained for $\mathbb{E}(v_t^{(j)})$. The algorithm then defines corrected estimates as

$$\boxed{\hat{m}_t^{(j)} = \frac{m_t^{(j)}}{1-\beta_1^t}}, \qquad \boxed{\hat{v}_t^{(j)} = \frac{v_t^{(j)}}{1-\beta_2^t}}$$

The update step for the j-th component of the weight vector is

$$\boxed{w_t^{(j)} = w_{t-1}^{(j)} - \gamma\frac{\hat{m}_t^{(j)}}{\sqrt{\hat{v}_t^{(j)}}+\varepsilon}}$$ (352)

[Note: Typical values for the hyperparameters are $\gamma = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$.]

XV- Normalisation

For reasons to present later, one may want to normalise the features $\{x_i\}$ before using them. There are many kinds of normalisation.

Normalisation (or min-max normalisation)

The min-max normalisation corresponds to negatively offset the components of the feature by the minimum value of that component and scale the result by the reciprocal of the difference between the maximum and the minimum, that is,

$$\boxed{z^{(j)} = \frac{x^{(j)} - x_{min}^{(j)}}{x_{max}^{(j)} - x_{min}^{(j)}}}$$ (588)

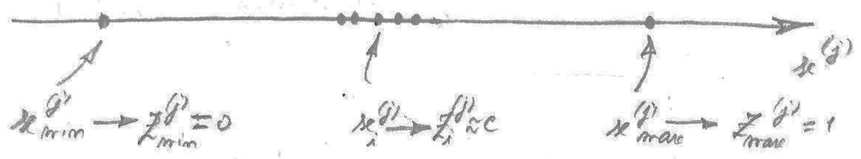
Or, in the vectorial form:

$$\boxed{\{z\} = (\{x\} - \{x_{min}\}) \oslash (\{x_{max}\} - \{x_{min}\})},$$ (589)

where the operator $\oslash$ is the Hadamard division. The min-max normalisation of the feature vector $\{x\}$ would make the resultant vector to be within the range $[0,1]^D$. Indeed,

$$z_{min}^{(j)} \leq z^{(j)} \leq z_{max}^{(j)} \Rightarrow \frac{x_{min}^{(j)} - x_{min}^{(j)}}{x_{max}^{(j)} - x_{min}^{(j)}} \leq z \leq \frac{x_{max}^{(j)} - x_{min}^{(j)}}{x_{max}^{(j)} - x_{min}^{(j)}} \Rightarrow 0 \leq z \leq 1.$$

Note, however, if there are some data points too far away from all the other ones, the min-max normalization may make the resultant components to assume very few different values. Consider the example below in which the extrema are transformed into 0 and 1 and the remainder are transformed into values very close to a constant $c \in]0,1[$:



Such transformation is not wanted. Standardization would rather be preferred. In fact, it is the usually the most appropriate type of normalization.

Standardization (or Z-score normalization)

Standardization is also termed Z-score normalization for historic reasons: the letter "Z" had been used to represent such transformation. In standardization the feature vectors $\{x_i\}_i$ are transformed so that the resultant means and standard deviations of the components are 0 and 1, respectively, that is, the ones of a standard normal distribution. The transformations may be expressed as

$$\boxed{z^{(j)} = \frac{x^{(j)} - \overline{x^{(j)}}}{s^{(j)}}}, \tag{590}$$

$$\boxed{\{z_i\} = (\{x_i\} - \overline{\{x_i\}}) \oslash \{s_j\}} \tag{591}$$

where

$$\boxed{\overline{x^{(j)}} = \frac{1}{m} \sum_{i=1}^{m} x_i^{(j)}} \tag{592}$$

and

$$\boxed{s^{(j)} = \sqrt{\frac{1}{m-1} \sum_{i=1}^{m} (x_i^{(j)} - \overline{x^{(j)}})^2}} \tag{593}$$

are the arithmetic mean and corrected sample standard deviation of the j-th component of the feature vectors, respectively (see expressions (93) and (102). Note that standard deviation is defined as the square root of the variance, that is,

$$\boxed{\sigma_x = \sqrt{\sigma_x^2} = \sqrt{\mathbb{E}(X - \mu)^2}} \tag{594}$$

One may show that $\overline{z^{(j)}} = 0$ and $s_{z^{(j)}} = 1$. Indeed,

$$\overline{z^{(j)}} = \frac{1}{m} \sum_{i=1}^{m} z_i^{(j)} \overset{(590)}{=} \frac{1}{m} \frac{\left(\sum_{i=1}^{m} x_i^{(j)}\right) - m\,\overline{x^{(j)}}}{s^{(j)}} = \frac{1}{m} \frac{m\,\overline{x^{(j)}} - m\,\overline{x^{(j)}}}{s^{(j)}} = 0 \quad \square$$

$$s_{z^{(j)}} = \sqrt{\frac{1}{m-1} \sum_{i=1}^{m} (z_i^{(j)} - \underbrace{\overline{z^{(j)}}}_{=0})^2} = \sqrt{\frac{1}{m-1} \sum_{i=1}^{m} z_i^{(j)}} = \sqrt{\frac{m}{m-1} \underbrace{\overline{z^{(j)}}}_{=1}}$$

$$= \sqrt{\frac{m}{m-1}} \xrightarrow[m \to +\infty]{} 1 \quad \square$$

Importance of normalization of features for Gradient Descent

Feature normalization is important efficiently attain the minimum of a cost function through Gradient Descent. Consider the example of Linear Regression as seen previously — the contour line of $f(\{w\})$ of value $f(\{w^*\}) + \mathcal{C}$, where $f(\{w\})$ is the cost function, is defined by (423):

$$\boxed{(\{w\} - \{w^*\})^T [A][A]^T (\{w\} - \{w^*\}) = \mathcal{C}} \tag{595}$$

Equation (595) defines a hyperellipse centred at $\{w\} = \{w^*\}$. Consider the example of the feature vector $\{x\}$ being subtracted by its mean, $\{\mu\}$, and the components being independent from each other. In that case, as shown by (502) the product $[A][A]^T$ corresponds to the n-scaled covariance matrix, $n \cdot [\Sigma]$ and the non-diagonal entries are null. In such conditions, one could express $[A][A]^T$ as

$$\boxed{[A][A]^T = n \sum_{j=1}^{D} \sigma_j^2 \{e_j\}\{e_j\}^T} \tag{596}$$

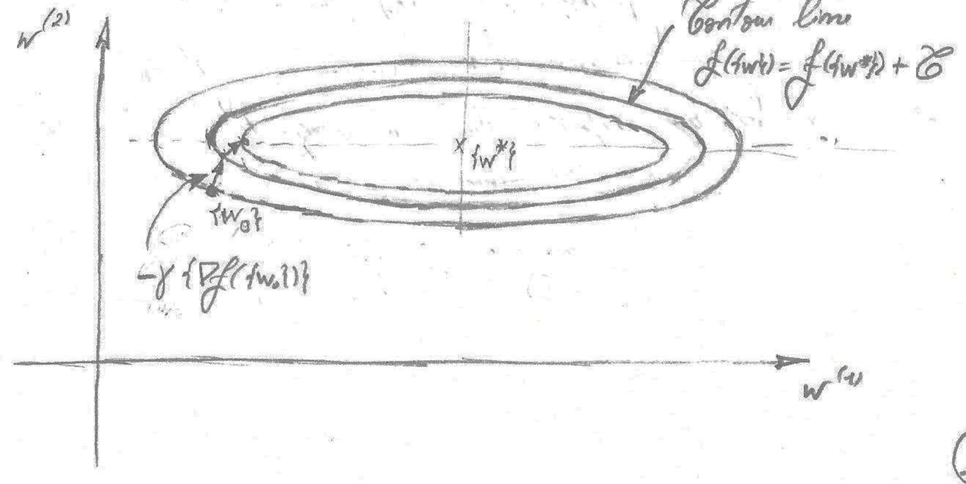
where $\sigma_j^2 = \mathbb{E}(x^{(j)} - \mu^{(j)})^2$ is the variance of the j-th component. Equation (595) then becomes

$$(\{w\} - \{w^*\})^T n \sum_{j=1}^{D} \sigma_j^2 \{e_j\}\{e_j\}^T (\{w\} - \{w^*\}) = \mathcal{C} \quad (=)$$

$$(=) \sum_{j=1}^{D} n\sigma_j^2 (w^{(j)} - w^{*(j)})^2 = \mathcal{C} \quad (=)$$

$$(=) \boxed{\sum_{j=1}^{D} \left(\frac{w^{(j)} - w^{*(j)}}{\sqrt{\frac{\mathcal{C}}{n\sigma_j^2}}} \right)^2 = 1} \tag{597}$$

Equation (597) defines a hyperellipse centred at $\{w\} = \{w^*\}$ and with semi-axes of lengths $\sqrt{\frac{\mathcal{C}}{n\sigma_j^2}}$, $j = 1, \dots, D$. The greatest semi-axes would have the smallest variance values, σ_j^2. If there are big discrepancies in the variances, then the hyperellipse would be close to flat. Descending from a point close to the vertex with the greatest semi-axis length to the minimum would be quite inefficient as shown in the picture below. If the feature components had comparable dimensions the hyperellipse would resemble a hypersphere making the descent to be more efficient. This shows how important normalization is for the application of Gradient Descent.

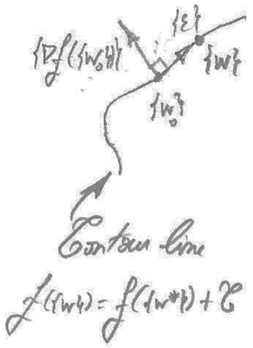


Note: (Representation of $\{\nabla f\}$ in a contour line)

Let one consider the case $D=2$, and the contour line defined by $\mathcal{L}(\{w\}) = f(\{w^*\}) + C$ where C is a constant. The gradient of f at some point $\{w_0\}$ of the line corresponds to

$$\{\nabla f(\{w_0\})\} = \frac{\partial f}{\partial w^{(1)}}(\{w_0\})\{e_1\} + \frac{\partial f}{\partial w^{(2)}}(\{w_0\})\{e_2\}.$$

Along the line the value of $\mathcal{L}(\{w\})$ does not change. Therefore,



$$\mathcal{L}(\{w\}) = \mathcal{L}(\{w_0\}) \Leftrightarrow f(\{w_0\}+\{\varepsilon\}) - f(\{w_0\}) = 0 \Leftrightarrow$$

$$\Leftrightarrow f(\{w_0\}) + \{\nabla f(\{w_0\})\}^T\{\varepsilon\} + O(\|\{\varepsilon\}\|^2) - f(\{w_0\}) = 0 \Leftrightarrow$$

$$\Leftrightarrow \lim_{\|\{\varepsilon\}\|\to 0} \{\nabla f(\{w_0\})\}^T\{\varepsilon\} = 0$$

Since when $\|\{\varepsilon\}\| \to 0$, $\{\varepsilon\}$ is tangent to the contour line, one knows that $\{\nabla f(\{w_0\})\}^T$ and the line at $\{w_0\}$ are orthogonal. Furthermore, from the definition of the gradient, $\{\nabla f(\{w_0\})\}^T$ is a vector that lies in the plane of vectors $\{w\}$. Also, from the sign of the components of the gradient, one concludes that the gradient points in the orthogonal direction for which there is an increase of the function value.