A theoretical framework for analysing the dynamics of LVQ
Ghosh, A.; Biehl, M.; Freking, A.; Reents, G.

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Final author's version (accepted by publisher, after peer review)

Publication date:
2004

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.
A theoretical framework for analyzing the dynamics of learning vector quantization: A statistical physics approach

Anarta Ghosh\(^1\), Michael Biehl\(^1\), Ansgar Freking\(^2\), and Georg Reents\(^2\) *

\(^1\) Institute of Mathematics and Computing Science, University of Groningen. P.O.Box. 800, 9700 AV Groningen, The Netherlands.
\(^2\) Institut für Theoretische Physik und Astrophysik, Julius-Maximilians-Universität Würzburg Am Hubland, 97074 Würzburg, Germany

1 Introduction

Concepts from statistical physics have been applied with success in the theory of learning. To investigate the typical behavior of large systems an averaging is performed over the disorder brought in by the stochastic nature of the input training data. A detailed overview of the use of methods from statistical physics in the field of machine learning can be found in [11], [12]. In this report we present a detailed derivation of a framework, employed in studying the dynamics of a class of supervised on-line learning algorithm called Learning vector quantization (LVQ), which is used to learn the prototype vectors for nearest prototype classification (NPC).

The NPC is perhaps the simplest classifier in pattern recognition [4]. Let the integer \(\tilde{c} > 1\) denote the number of classes in a labeled dataset. Let \(\Omega = \{w_1, \ldots, w_c\} \subset \mathbb{R}^N, c \geq \tilde{c}\), be the set of prototypes, where each \(w_i \in \Omega\) is labeled with one of the \(\tilde{c}\) classes. The NPC assigns any unlabeled object \(\xi \in \mathbb{R}^N\) to the class of its nearest (with respect to some distance metric) prototype. The NPC can be regarded as a one-nearest-neighbor (1-NN) rule where the reference set is \(\Omega\). The subtle difference between the two paradigms is that NPC assumes a much smaller number of prototypes than the 1-NN rule, which makes it more computationally efficient. The main problem in building an NPC is to construct a good prototype set which will ideally be of minimal cardinality and has minimum generalization error.

LVQ, first proposed by Kohonen ([5]) is the class of supervised on-line learning algorithms employed to find the elements of \(\Omega\) which will "best" represent the classes. The number of classes(\(c\)) and the number of prototypes, (the cardinality of \(\Omega, c\)), are predefined. A labeled training dataset of the form \((\xi^\mu, \sigma^\mu)\) (where \(\xi^\mu \in \mathbb{R}^N\) is the training data vector presented to the on-line learning

algorithm at time stamp $\mu$ and $\sigma^\mu$ is its class label) is used to learn the elements of $\Omega$. Various modifications of the original LVQ algorithms proposed in [3] exist which ensure for example faster convergence, better approximation of optimal (Bayesian) classification error, better choice and management of dimensionality of the training data ([5], [7], [9], [6], [8], [10]). The generic structure of an LVQ algorithm can be expressed in the following way:

$$w_l^\mu = w_l^{\mu-1} + \Delta w_l^\mu,$$

$$= w_l^{\mu-1} + \frac{\eta}{N} f(\{w_l^{\mu-1}\}, \xi^\mu, \sigma^\mu)(\xi^\mu - w_i^\mu), l = 1 \ldots c, \mu = 1, 2 \ldots (1)$$

Where $\eta$ is the "so called" learning rate, $N$ is the dimensionality of the system. The specific form of $f$ is determined by the algorithm used to perform LVQ.

Statistical physics provides the tools to study the stability and convergence of such on-line learning scenario described by expression (1) in the limit of infinite dimensionality ($N \to \infty$). The schema of such an analysis is as follows:

- Define the order parameters in terms of which the system dynamics will be analyzed. The number of order parameters is typically much less than the actual dimensionality of the system ($N$).
- Derive the recurrence relations for order parameter from the learning algorithm similar to the generic expression (1).
- In the thermodynamic limit ($N \to \infty$) the recurrence relations yield coupled system of differential equations and the order parameters become self averaging [1] with respect to the stochastic sequence of input training data.
- We study the dynamics of the system by solving the afore mentioned system of differential equations.

In this report we present the detailed derivation of the above mentioned coupled system of differential equations for three different classes of LVQ algorithms which are described in Section 2. We consider a simple two class problem where the number of prototypes to be learnt is also two and the classes are labelled as $\pm 1$.

In Section 2 we present the model, definitions and a brief description of the algorithms we intend to analyze. Detailed derivation of the differential equations is presented in Section 3. In Section 4 we discuss some further aspects of the proposed framework. We conclude with a brief summary is Section 5.

2 The model, notations, definitions and LVQ algorithms

2.1 The model

Prototypes: $w_s \in \mathbb{R}^N, s \in \{+1, -1\}$ - the prototype vectors for two different classes.
**Data Model:** \((\xi^\mu, \sigma^\mu)\) is the model of the data. \(\xi^\mu\) is the input data vector. \(\mu\) is the time stamp at which the data is observed, \(\mu = 1, 2, \ldots\). \(\sigma^\mu\) is the class to which the observed data \(\xi^\mu\) belongs to at time step \(\mu\), \(\sigma^\mu \in \{+1, -1\}\). We assume that \(\xi^\mu\) is independently drawn from a dataset having a probability density function \(p(\xi)\) which is defined as follows:

\[
p(\xi) = \sum_{\sigma = \pm 1} p_\sigma p(\xi|\sigma),
\]

where

\[
p_\sigma = P(\sigma = \pm 1) \quad \text{and} \quad p(\xi|\sigma) \quad \text{is the conditional probability density of the data given the class} \; \sigma.
\]

In our study we choose \(p(\xi|\sigma)\) to be normally distributed with independent components (the covariance matrix is diagonal), i.e.

\[
p(\xi|\sigma) = \frac{1}{(2\pi v_\sigma)^{\frac{N}{2}}} \exp\left[ -\frac{1}{2} \frac{(\xi - \lambda B_\sigma)^2}{v_\sigma} \right].
\]

Where \(\lambda\) is the magnitude of the mean vector, \(\|B_\sigma\| = 1\), \(B_\sigma\) specifies the direction of the mean vector for class \(\sigma\). For any \(x \in \mathbb{R}^N\), \(x^2 \equiv x \cdot x\); \(\cdot\) denotes scalar product. \(v_1\) and \(v_{-1}\) are the variances of each component of the data vector \(\xi\) corresponding to class labels \(\sigma = 1\) and \(\sigma = -1\) respectively. \(< . . >\) denotes average (expectation) over \(p(\xi)\) and can be expressed in the following form:

\[
< . . > = \sum_{\sigma = \pm 1} p_\sigma < . . >_\sigma
\]

Where \(< . . >_\sigma\) is the conditional average for class \(\sigma\). We choose the input data, \(\xi\), in such a way that,

\[
< \xi \cdot \xi > = \sum_{\sigma = \pm 1} p_\sigma < \xi^2 >_\sigma
\]

\[
= p_1(Nv_1 + \lambda^2) + p_{-1}(Nv_{-1} + \lambda^2)
\]

\[
\approx N(p_1v_1 + p_{-1}v_{-1})
\]

\(\langle \cdot . N > > \lambda, < \xi^2 >_\sigma \approx Nv_\sigma\)

### 2.2 Order Parameters and Projections

We define the order parameters \((R_{lm}, Q_{lm})\) and the projections \((b_l, h_l)\) as follows:

\[
R_{lm} = w_l \cdot B_m
\]

\[
Q_{lm} = w_l \cdot w_m
\]

\[
b_l = \xi \cdot B_l
\]

\[
h_l = \xi \cdot w_l
\]

Define,

\[
x = (h_1, h_{-1}, b_1, b_{-1})
\]

Instantaneous projections: \(b_l^\mu = \xi^\mu \cdot B_l\); \(h_l^\mu = w_l^{\mu-1} \cdot \xi^\mu\)

Instantaneous order parameters: \(R_{lm}^\mu = w_l^\mu \cdot B_m\); \(Q_{lm}^\mu = w_l^\mu \cdot w_m\)

Throughout this article \(l, m, k \in \{+1, -1\}\).
2.3 Statistics of the Projections

Given that each training vector is independent of all previous ones, the statistical properties of the projections are well defined for large $N$. Central limit theorem yields that their joint density, $p(h_{+1}, h_{-1}, b_{+1}, b_{-1}) = p(x)$, is normally distributed and fully specified by the corresponding conditional averages and covariances.

First Order Statistics of $h$:

$$< h_l >_k = \int_{\mathbb{R}^N} \xi \cdot w_l p(\xi | \sigma = k) d\xi = w_l \cdot \int_{\mathbb{R}^N} \xi p(\xi | \sigma = k) d\xi$$

$$= w_{l} \cdot \lambda B_k = \lambda R_{lk}$$  \hspace{1cm} (8)

First Order Statistics of $b$:

$$< b_l >_k = \int_{\mathbb{R}^N} \xi \cdot B_l(\xi | \sigma = k) d\xi = B_l \cdot \int_{\mathbb{R}^N} \xi p(\xi | \sigma = k) d\xi$$

$$= B_l \cdot \lambda B_k = \lambda T_{lk} \text{(say)} \hspace{1cm} (9)$$

Where $T_{lk} \equiv B_l \cdot B_k$ is called the overlap parameter.

Hence the conditional means of $x$ for two classes can be expressed in the following way:

$$\mu_k = \begin{pmatrix} \lambda R_{1,1} \\ \lambda R_{1,-1} \\ \lambda T_{1,1} \\ \lambda T_{1,-1} \end{pmatrix}$$

Second Order Statistics of $h$:

To compute the conditional variance let us first look at the average,

$$< h_l h_m >_k = < (w_l \cdot \xi)(w_m \cdot \xi) >_k$$

$$= \left\langle \sum_{i=1}^{N} (w_l)_i (\xi)_i \sum_{j=1}^{N} (w_m)_j (\xi)_j \right\rangle_k$$

$$= \left\langle \sum_{i=1}^{N} (w_l)_i (w_m)_i (\xi)_i (\xi)_i + \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} (w_l)_i (w_m)_j (\xi)_i (\xi)_j \right\rangle_k$$

$$= \sum_{i=1}^{N} (w_l)_i (w_m)_i < (\xi)_i (\xi)_i >_k + \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} (w_l)_i (w_m)_j < (\xi)_i (\xi)_j >_k$$

$$= \sum_{i=1}^{N} (w_l)_i (w_m)_i \left[ \nu_k + \lambda^2 (B_k)_i (B_k)_i \right]$$
\[
\sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} (w_l)_i (w_m)_j \lambda^2 (B_k)_i (B_k)_j
\]

\[
\therefore \text{components of } \xi \text{ have variance } v_k,
\]

\[
\Rightarrow \langle (\xi)_i, (\xi)_i \rangle > k - \langle (\xi)_i \rangle > k < (\xi)_i > k = v_k, \forall i \in \{1 \ldots N\}
\]

\[
\Rightarrow \langle (\xi)_i, (\xi)_j \rangle > k = v_k + \langle (\xi)_i \rangle > k < (\xi)_i > k
\]

and they are independent

\[
\Rightarrow \langle (\xi)_i, (\xi)_j \rangle > k - \langle (\xi)_i \rangle > k < (\xi)_j > k = 0, \forall i, j \in \{1 \ldots N\}, i \neq j.
\]

\[
\Rightarrow \langle (\xi)_i, (\xi)_j \rangle > k = \langle (\xi)_i \rangle > k < (\xi)_j > k
\]

\[
= v_k \sum_{i=1}^{N} (w_l)_i (w_m)_i + \lambda^2 \sum_{i=1}^{N} (w_l)_i (w_m)_i (B_k)_i (B_k)_i
\]

\[
+ \lambda^2 \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} (w_l)_i (w_m)_j (B_k)_i (B_k)_j
\]

\[
= v_k w_l \cdot w_m + \lambda^2 (w_l \cdot B_k)(w_m \cdot B_k)
\]

\[
= v_k Q_{lm} + \lambda^2 R_{lk} R_{mk}
\]

Hence we have,

\[
\langle h_l, h_m \rangle > k - \langle h_l \rangle > k < h_m > k = v_k Q_{lm} + \lambda^2 R_{lk} R_{mk} - \lambda^2 R_{lk} R_{mk}
\]

\[
= v_k Q_{lm}
\]

(11)

Second Order Statistics of \( b \): Similar to (12) we get the second order statistics for \( b \) as follows:

\[
\langle b_l b_m \rangle > k - \langle b_l \rangle > k < b_m > k = v_k T_{lm} + \lambda^2 T_{lk} T_{mk} - \lambda^2 T_{lk} T_{mk} = v_k T_{lm}
\]

(13)

Covariance: \( \langle h_l b_m \rangle > k - \langle h_l \rangle > k < b_m > k \)

To compute the conditional variance let us first look at the average,

\[
\langle h_l b_m \rangle > k = \langle (w_l \cdot \xi)(B_m \cdot \xi) \rangle > k
\]

\[
= v_k w_l \cdot B_m + \lambda^2 (w_l \cdot B_k)(B_m \cdot B_k)
\]

\[
= v_k R_{lm} + \lambda^2 R_{lk} T_{mk}
\]

(14)

Hence we have,

\[
\langle h_l, b_m \rangle > k - \langle h_l \rangle > k < b_m > k = v_k R_{lm} + \lambda^2 R_{lk} T_{mk} - \lambda^2 R_{lk} T_{mk}
\]

\[
= v_k R_{lm}
\]

(15)
The conditional density of $x$ for class $k$ is $N(\mu_k, C_k)$, where, $\mu_k$ is the conditional mean vector for class $k$ and $C_k$ is the conditional covariance matrix for class $k$. In our study the conditional density of $x$ is a 4-dimensional Gaussian. The covariance matrix by $C_k$ and can be explicitly expressed as follows:

$$C_k = v_k \begin{pmatrix} Q_{1,1} & Q_{1,-1} & R_{1,1} & R_{1,-1} \\ Q_{1,-1} & Q_{-1,1} & R_{-1,1} & R_{-1,-1} \\ R_{1,1} & R_{-1,1} & T_{1,1} & T_{1,-1} \\ R_{1,-1} & R_{-1,-1} & T_{1,-1} & T_{-1,-1} \end{pmatrix}$$ (16)

2.4 LVQ algorithms

In this article we present a detailed derivation of the system of coupled differential equations in terms of order parameters, used in the dynamical analysis of the following LVQ algorithms.

LVQ2.1: LVQ2.1 has been shown to provide good NPC classifiers ([14], [15]). For every data point $(\xi^\mu, \sigma^\mu)$, LVQ2.1 first selects two nearest prototypes according to the Euclidean distance metric. If labels of such two nearest prototypes are different and one of them is $\sigma^\mu$, then the prototypes are updated. In the model (two prototypes and two classes of training data) we study the LVQ2.1 algorithm can be described in the following way:

$$w_{l^\mu} = w_{l^\mu}^{\mu-1} + (\Delta w_{l^\mu})_{LVQ2.1}$$

$$= w_{l^\mu}^{\mu-1} + \eta N (l(\sigma^\mu)(\xi^\mu - w_{l^\mu}^{\mu-1}) \quad \text{(17)}$$

Robust soft learning vector quantization (RSLVQ): In this report we deal with the "hard" version of RSLVQ proposed in [2]. For a given training data point $(\xi^\mu, \sigma^\mu)$, this LVQ algorithm determines the nearest prototype, say $w_l$ and the nearest prototype, $w_{\sigma^\mu}$ which has the same class label as the data class label, $\sigma^\mu$ using Euclidean distance metric. If $l = \sigma^\mu$ then no update of the prototypes is performed, otherwise the prototypes are updated. In the model we use this algorithm updates the prototypes when the winner (nearest prototype) has a different class label from the class label of the input data in the following way:

$$w_{l^\mu} = w_{l^\mu}^{\mu-1} + (\Delta w_{l^\mu})_{RSLVQ}$$

$$= w_{l^\mu}^{\mu-1} + \frac{\eta}{N} (l(\sigma^\mu)(\xi^\mu - w_{l^\mu}^{\mu-1}) \Theta (d_{+\sigma^\mu} - d_{-\sigma^\mu}) \quad \text{(18)}$$

Where, $\Theta(.)$ is the Heaviside function, i.e.

$$\Theta(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

$$d_{\sigma^\mu} = (\xi^\mu - w_{\sigma^\mu}^{\mu-1})^2 \quad \text{(19)}$$

We denote $\Theta (d_{+\sigma^\mu} - d_{-\sigma^\mu})$ by $\Theta_{\sigma^\mu}$ in short.
**Winner takes all algorithms** In this LVQ algorithm for every training data point, \((\xi^\mu, \sigma^\mu)\), only (and always) the nearest prototype is updated according to the following prescription:

\[
\mathbf{w}_l^\mu = \mathbf{w}_l^{\mu-1} + \frac{\eta}{N} \left[ a + b \sigma^\mu \right] \Theta(d_{-l} - d_l)(\xi^\mu - \mathbf{w}_l^{\mu-1})
\]  

(20)

where \(l \in \{+1, -1\}\), \(a, b \in \mathbb{R}\). We denote \(\Theta(d_{-l} - d_l)\) by \(\Theta_l\) in short. The aforementioned competitive or winner takes all learning algorithm moves the nearest prototype (the winner) towards or away from the current input training data depending on the sign of \([a + b \sigma^\mu]\). By choosing different values for the parameters \(a\) and \(b\) we can interpolate between unsupervised vector quantization (VQ) and LVQ. Following three cases will be of special interest.

(I) \(a= 1, b= 0\) (VQ): The label of input training data point, \(\sigma^\mu\) does not play any role in the learning process and the winner is always moved towards the data point, \(\xi^\mu\). This case corresponds to the well-known unsupervised winner takes all paradigm ([13]).

The updates can be interpreted as a gradient descent step with respect to the cost function,

\[
E^\mu = \frac{1}{2} \sum_{l=\pm 1} \Theta(d_{-l}^\mu - d_l^\mu)(\xi^\mu - \mathbf{w}_l^{\mu-1})^2
\]  

(21)

(II) \(a= 0, b= 1\) (LVQ1): In this case the nearest prototype is moved closer to (away from) the input training at any instance if label of the nearest prototype is same (different) as (from) the class label of the input data. With these parameter values the expression (20) reproduces the original formulation of the LVQ, formally named as LVQ1.

(III) \(a=b= \frac{1}{2}\) (LVQ+): The nearest prototype is updated only if the class level of the nearest prototype is same as the class label of the input data, otherwise no update is made. Formally it is called LVQ+.

In the cases (II) and (III) the correct classification of the current example is taken into account. The update of the prototypes are designed to decrease the probability of misclassifying similar inputs in the future. These heuristic algorithms are compromises between the unsupervised detection of structures in the input data space and supervised learning of the classification scheme.

3 Detailed derivation of system of coupled differential equation

Under the assumption that at each step the afore mentioned algorithms are driven by the presentation of a single training vector which is independent of the previous ones, the evolution of the prototypes in this paradigm can be conceptually conceived as a stochastic process, to be precise a Markov process. If the
underlying distribution of the training data is simple enough, the whole dynamics of the system can be analyzed using a few order parameter \( \{ R_{lm}, Q_{lm} \} \) and in the thermodynamic limit \( (N \rightarrow \infty) \) these order parameters are self-averaging ([1]), i.e., the variances of the probability density functions of the order parameters vanish in this limit of infinite dimensionality. This self-averaging property of the order parameters allows us to analyze the stochastic evolution of the prototype vector in terms of deterministic evolution of order parameters, which greatly helps to build a theoretical understanding of such systems. This deterministic evolution of order parameters is mathematically described using a system of coupled differential equations. In the following subsections we present a detailed derivation of such system of coupled differential equations for the above mentioned (17, 18, 20) three different LVQ algorithms.

### 3.1 LVQ2.1

**Recurrence relations for order parameters \( R \):**

\[
R_{lm}^\mu = w_l^\mu \cdot B_m \\
= \left( w_l^{\mu-1} + (\Delta w_l^\mu)_{lvq2.1} \right) \cdot B_m \\
= w_l^{\mu-1} \cdot B_m + \left( (\Delta w_l^\mu)_{lvq2.1} \right) \cdot B_m \\
= R_{lm}^{\mu-1} + \frac{\eta}{N} (l\sigma^\mu) \left( b_m^\mu - R_{lm}^{\mu-1} \right) \tag{22}
\]

**Recurrence relation for order Parameter \( Q \):**

\[
Q_{lm}^\mu = w_l^\mu \cdot w_m^\mu \\
= \left( w_l^{\mu-1} + (\Delta w_l^\mu)_{lvq2.1} \right) \cdot \left( w_m^{\mu-1} + (\Delta w_m^\mu)_{lvq2.1} \right) \\
= Q_{lm}^{\mu-1} + w_l^{\mu-1} \cdot (\Delta w_m^\mu)_{lvq2.1} + (\Delta w_l^\mu)_{lvq2.1} \cdot w_m^{\mu-1} \\
+ (\Delta w_l^\mu)_{lvq2.1} \cdot (\Delta w_m^\mu)_{lvq2.1} \\
= Q_{lm}^{\mu-1} + \frac{\eta}{N} \left( l\sigma^\mu h_m^{\mu-1} - l\sigma^\mu Q_{lm}^{\mu-1} + m\sigma^\mu h_l^{\mu-1} - m\sigma^\mu Q_{lm}^{\mu-1} \right) \\
+ \frac{\eta}{N} (l\sigma^\mu) (l\sigma^\mu) \tag{23}
\]

Since the analysis is for very large \( N \) we neglect the terms of \( O\left(\frac{1}{N^2}\right) \) in (23).
**Differential equations** In the thermodynamic limit, i.e. $N \rightarrow \infty$, the variable $\alpha = \frac{\mu}{N}$ is a continuous one and $d\alpha = \frac{d\mu}{N} = \frac{1}{N}$. Hence in the thermodynamic limit from the recurrence relation in (22) we get,

$$\frac{dR_{lm}}{d\alpha} = \eta(l) \left( \langle \sigma b_m \rangle - \langle \sigma \rangle R_{lm} \right)$$

(24)

Similarly to (24) for $Q$ we have,

$$\frac{dQ_{lm}}{d\alpha} = \eta \left( l \langle \sigma h_m \rangle - l \langle \sigma \rangle Q_{lm} + m \langle \sigma \rangle Q_{lm} + \eta(lm)(p_1 v_1 + p_{-1} v_{-1}) \right) \right)$$

(25)

From (24) and (25) we see that we need to calculate the following averages:

A. $\langle \sigma b_m \rangle = \sum_{\sigma=\pm 1} \int_{R\cdot N} \sigma b_m \sigma \lambda T_{m,\sigma} d\xi$

B. $\langle \sigma h_m \rangle = \sum_{\sigma=\pm 1} \int_{R\cdot N} \sigma h_m \sigma \lambda R_{m,\sigma} d\xi$

C. $\langle \sigma \rangle = \sum_{\sigma=\pm 1} \sigma \rho \sigma \lambda R_{m,\sigma}$

Equation (24) and (25) along with expressions for averages above imply the following differential equations which are needed to be solved to analyze the dynamics of LVQ2.1 algorithm.

$$\frac{dR_{lm}}{d\alpha} = \eta(l) \left( \sum_{\sigma=\pm 1} \sigma \rho \sigma \lambda T_{m,\sigma} - \sum_{\sigma=\pm 1} \sigma \rho \sigma R_{lm} \right)$$

(26)

$$\frac{dQ_{lm}}{d\alpha} = \eta \left( l \sum_{\sigma=\pm 1} \sigma \rho \sigma \lambda R_{m,\sigma} - l \sum_{\sigma=\pm 1} \sigma \rho \sigma Q_{lm} + m \sum_{\sigma=\pm 1} \sigma \rho \sigma \lambda R_{m,\sigma} - m \sum_{\sigma=\pm 1} \sigma \rho \sigma Q_{lm} + \eta(lm)(p_1 v_1 + p_{-1} v_{-1}) \right)$$

(27)

### 3.2 RSLVQ

Recurrence relation for order parameter $R$:

$$R_{lm} = w_1^{\alpha} \cdot B_m$$
\[
\begin{align*}
&= \left( w_{\mu}^{\mu-1} + (\Delta w_{\mu})_{\text{RSLVQ}} \right) \cdot B_m \\
&= R_{\mu}^{\mu-1} + \frac{\eta}{N} (l\sigma^\mu)(h_{\sigma^\mu} - R_{\mu}^{\mu-1}{\Theta_{\sigma^\mu}})
\end{align*}
\] (28)

Recurrence relation for order parameter Q:

\[
Q_{\mu \sigma}^{\mu} = w_{\mu}^{\mu} \cdot w_{m}^{\mu}
\]
\[
= \left( w_{\mu}^{\mu-1} + (\Delta w_{\mu})_{\text{RSLVQ}} \right) \cdot \left( w_{m}^{\mu-1} + (\Delta w_{m}^{\mu})_{\text{RSLVQ}} \right)
\]
\[
= Q_{\mu \sigma}^{\mu-1} + \frac{\eta}{N} \left( l\sigma^\mu h_{\mu}^{\mu-1} - l\sigma^\mu Q_{\mu \sigma}^{\mu-1} + m\sigma^\mu h_{\mu}^{\mu-1} - m\sigma^\mu Q_{\mu \sigma}^{\mu-1} \right)
\]
\[
+ \eta (l\mu) \xi \cdot \xi \Theta_{\sigma^\mu}
\] (29)

We neglect the terms of \(O(\frac{1}{N})\) in (29).

Differential equations In the thermodynamic limit (28) and (29) becomes the following differential equations:

\[
\frac{dR_{\mu \sigma}^{\mu}}{d\alpha} = \eta(l) \left( < \sigma b_{\mu \sigma} \Theta_{\sigma} > - < \sigma \Theta_{\sigma} > R_{\mu \sigma}^{\mu} \right) 
\] (30)

\[
\frac{dQ_{\mu \sigma}^{\mu}}{d\alpha} = \eta \left( l < \sigma h_{\mu} \Theta_{\sigma} > - l < \sigma \Theta_{\sigma} > Q_{\mu \sigma} + m < \sigma h_{\sigma} \Theta_{\sigma} > \\
- m < \sigma \Theta_{\sigma} > Q_{\mu \sigma} \\
+ (l\mu)\eta(p_1 v_1 < \Theta_{\sigma} >_1 + p_{-1} v_{-1} < \Theta_{\sigma} >_{-1}) \right)
\] (31)

To compute the averages in the differential equations above (30,31) let us look at the function \(\Theta_{\sigma}\).

\[
\Theta_{\sigma} = \Theta(d_{+\sigma} - d_{-\sigma})
\]
\[
= \Theta \left( (\xi - w_{+\sigma})^2 - (\xi - w_{-\sigma})^2 \right)
\]
\[
= \Theta \left( -2\xi \cdot w_{+\sigma} + w_{+\sigma} \cdot w_{+\sigma} + 2\xi \cdot w_{-\sigma} - w_{-\sigma} \cdot w_{-\sigma} \right)
\]
\[
= \Theta \left( -2h_{+\sigma} + Q_{+\sigma,+\sigma} + 2h_{-\sigma} - Q_{-\sigma,-\sigma} \right)
\]
\[
\Theta = \Theta \left( -2, +2, 0, 0 \cdot (h_{+\sigma}, h_{-\sigma}, b_{+\sigma}, b_{-\sigma}) + Q_{+\sigma,+\sigma} - Q_{-\sigma,-\sigma} \right)
\]

(32)

Hence we have,

\[
\Theta_{\sigma} = \Theta (\alpha_{\sigma} \cdot x - \beta_{\sigma}). \tag{33}
\]

Where, \(\alpha_{\sigma} = (-2\sigma, 2\sigma, 0, 0)\) and \(\beta_{\sigma} = -(Q_{\sigma}, Q_{-\sigma}, -Q_{-\sigma}, Q_{+\sigma})\).

As all the averages in (30) and (31) can be expressed as sum of weighted conditional averages, c.f. identity (3), it is enough to compute the averages of the following forms: \(< (x)_n \Theta_{\sigma} >_\sigma >_\sigma >_\sigma\), and \(< \Theta_{\sigma} >_\sigma \), where, \((x)_n\) is the \(n^{th}\) component of the vector \(x\); \(n \in \{1, 2, 3, 4\}\).

From expression (65) in the Appendix A we get the final form of the system of coupled differential equation as follows:

\[
\begin{align*}
\frac{dR_{lm}}{d\alpha} &= \eta \left( l \sum_{\sigma = \pm 1} \sigma p_{\sigma} \left[ \left( C_{\sigma} \alpha_{\sigma} \right)_{n_{bm}} \sqrt{2\pi} \tilde{\alpha}_{\sigma, \sigma} \exp \left[ -\frac{1}{2} \left( \tilde{\beta}_{\sigma, \sigma} \right)^2 \right] \right] + (\mu_{\sigma})_{n_{hm}} \phi \left( \frac{\tilde{\beta}_{\sigma, \sigma}}{\tilde{\alpha}_{\sigma, \sigma}} \right) R_{lm} \right) \\
\frac{dQ_{lm}}{d\alpha} &= \eta \left( l \sum_{\sigma = \pm 1} \sigma p_{\sigma} \left[ \left( C_{\sigma} \alpha_{\sigma} \right)_{n_{bm}} \sqrt{2\pi} \tilde{\alpha}_{\sigma, \sigma} \exp \left[ -\frac{1}{2} \left( \tilde{\beta}_{\sigma, \sigma} \right)^2 \right] + (\mu_{\sigma})_{n_{hm}} \phi \left( \frac{\tilde{\beta}_{\sigma, \sigma}}{\tilde{\alpha}_{\sigma, \sigma}} \right) \right] \\
&\quad - l \sum_{\sigma = \pm 1} \sigma p_{\sigma} \left[ \left( C_{\sigma} \alpha_{\sigma} \right)_{n_{hm}} \sqrt{2\pi} \tilde{\alpha}_{\sigma, \sigma} \exp \left[ -\frac{1}{2} \left( \tilde{\beta}_{\sigma, \sigma} \right)^2 \right] + (\mu_{\sigma})_{n_{hm}} \phi \left( \frac{\tilde{\beta}_{\sigma, \sigma}}{\tilde{\alpha}_{\sigma, \sigma}} \right) \right] \\
&\quad + m \sum_{\sigma = \pm 1} \sigma p_{\sigma} \left[ \left( C_{\sigma} \alpha_{\sigma} \right)_{n_{hm}} \sqrt{2\pi} \tilde{\alpha}_{\sigma, \sigma} \exp \left[ -\frac{1}{2} \left( \tilde{\beta}_{\sigma, \sigma} \right)^2 \right] + (\mu_{\sigma})_{n_{hm}} \phi \left( \frac{\tilde{\beta}_{\sigma, \sigma}}{\tilde{\alpha}_{\sigma, \sigma}} \right) \right] \\
&\quad - n \sum_{\sigma = \pm 1} \sigma p_{\sigma} \left[ \left( C_{\sigma} \alpha_{\sigma} \right)_{n_{hm}} \sqrt{2\pi} \tilde{\alpha}_{\sigma, \sigma} \exp \left[ -\frac{1}{2} \left( \tilde{\beta}_{\sigma, \sigma} \right)^2 \right] + (\mu_{\sigma})_{n_{hm}} \phi \left( \frac{\tilde{\beta}_{\sigma, \sigma}}{\tilde{\alpha}_{\sigma, \sigma}} \right) \right] \right) 
\end{align*}
\]

(35)

Where,

\[
\begin{align*}
n_{bm} &= 3 \text{ if } m = 1 \\
&= 4 \text{ if } m = -1 \\
n_{hm} &= 1 \text{ if } m = 1 \\
&= 2 \text{ if } m = -1 \\
\tilde{\alpha}_{\sigma} &= \sqrt{\alpha_{\sigma} C_{\sigma} \alpha_{\sigma}}, \quad \tilde{\beta}_{\sigma, \sigma} = \alpha_{\sigma} \cdot \mu_{\sigma} - \beta_{\sigma}
\end{align*}
\]

3.3 Winner takes all algorithms

Following exactly similar steps as in LVQ2.1 and RSLVQ we can express the required differential equations as follows:

\[
\frac{dR_{lm}}{d\alpha} = \eta a \left( < b_m \Theta_l > - < \Theta_l > R_{lm} \right)
\]
\[ dQ_{lm} = \eta \left[ b \left( l < h_{lm} \Theta_l > - l < \sigma \Theta_l > Q_{lm} + m < \sigma h_{lm} \Theta_m > 
- m < \sigma \Theta_m > Q_{lm} \right) 
+ a \left( < h_m \Theta_l > - < \Theta_l > Q_{lm} + < h_l \Theta_m > - < \Theta_m > Q_{lm} \right) 
+ \delta_{lm} \eta [a^2 + b^2 l m] (p_1 v_1 < \Theta_l >_1 + p_{-1} v_{-1} < \Theta_l >_{-1}) 
+ \delta_{lm} a b (l + m) (p_1 v_1 < \Theta_l >_1 + p_{-1} v_{-1} < \Theta_l >_{-1}) \right] \] \] (37)

Where,
\[ \delta_{lm} = 1 \text{ if } m = l 
= 0 \text{ if } m \neq l \]

Similar to \( \Theta_\sigma \) of RSLVQ we can express \( \Theta_l \) in the following way,
\[ \Theta_l = \Theta (\alpha_l \cdot x - \beta_l) \] (38)

Where \( \alpha_l = (2l, -2l, 0, 0) \) and \( \beta_l = (Q_{l,l} - Q_{l,-l}) \). To compute the averages in (36) and (37) it is enough to compute the averages of the following forms:
\( < (x)_n \Theta_l >_k \) and \( < \Theta_l >_k \). Using averages in (65) in Appendix A we get the following expression for differential equations,
\[ \frac{dR_{lm}}{d\alpha} = \eta \left[ b l \left( \sum_{\sigma = \pm 1} \sigma p_{\sigma} \left[ \frac{(C_\sigma \alpha_l)_{nm}}{\sqrt{2 \pi \alpha_{l,\sigma}}} \exp \left[ - \frac{1}{2} \left( \frac{\tilde{\beta}_{l,\sigma}}{\alpha_{l,\sigma}} \right)^2 \right] \right. \right. 
+ \left. \left. (\mu_\sigma)_{nm} \Phi \left( \frac{\tilde{\beta}_{l,\sigma}}{\alpha_{l,\sigma}} \right) \right] - \sum_{\sigma = \pm 1} \sigma p_{\sigma} \Phi \left( \frac{\tilde{\beta}_{l,\sigma}}{\alpha_{l,\sigma}} \right) R_{lm} \right) 
\] 
\[ + (a) \left( \sum_{\sigma = \pm 1} \sigma p_{\sigma} \left[ \frac{(C_\sigma \alpha_l)_{nm}}{\sqrt{2 \pi \alpha_{l,\sigma}}} \exp \left[ - \frac{1}{2} \left( \frac{\tilde{\beta}_{l,\sigma}}{\alpha_{l,\sigma}} \right)^2 \right] \right. \right. 
+ \left. \left. (\mu_\sigma)_{nm} \Phi \left( \frac{\tilde{\beta}_{l,\sigma}}{\alpha_{l,\sigma}} \right) \right] - \sum_{\sigma = \pm 1} \sigma p_{\sigma} \Phi \left( \frac{\tilde{\beta}_{l,\sigma}}{\alpha_{l,\sigma}} \right) R_{lm} \right) \] \] (39)

\[ \frac{dQ_{lm}}{d\alpha} = \eta \left[ b l \sum_{\sigma = \pm 1} \sigma p_{\sigma} \left[ \frac{(C_\sigma \alpha_l)_{nm}}{\sqrt{2 \pi \alpha_{l,\sigma}}} \exp \left[ - \frac{1}{2} \left( \frac{\tilde{\beta}_{l,\sigma}}{\alpha_{l,\sigma}} \right)^2 \right] \right. \right. 
+ \left. \left. (\mu_\sigma)_{nm} \Phi \left( \frac{\tilde{\beta}_{l,\sigma}}{\alpha_{l,\sigma}} \right) \right] - b l \sum_{\sigma = \pm 1} \sigma p_{\sigma} \Phi \left( \frac{\tilde{\beta}_{l,\sigma}}{\alpha_{l,\sigma}} \right) Q_{lm} \right] \]
+ \sum_{\sigma=\pm 1} \sigma p_{\sigma} \left[ \frac{(C_{\sigma} \alpha_l)_{nm}}{\sqrt{2\pi \tilde{\alpha}_{m,\sigma}}} \exp \left[ -\frac{1}{2} \left( \frac{\tilde{\beta}_{m,\sigma}}{\tilde{\alpha}_{m,\sigma}} \right)^2 \right] \right]

- \sum_{\sigma=\pm 1} \sigma p_{\sigma} \left[ \frac{\phi \left( \frac{\tilde{\beta}_{m,\sigma}}{\tilde{\alpha}_{m,\sigma}} \right)}{Q_{lm}} \right]

+ \delta_{lm} (a^2 + b^2 lm) \eta^2 \sum_{\sigma=\pm 1} \sigma p_{\sigma} v_{\sigma} \phi \left( \frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}} \right)

+ \delta_{lm} \eta^2 (ab(l + m)) \sum_{\sigma=\pm 1} \sigma p_{\sigma} v_{\sigma} \phi \left( \frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}} \right)

+ \eta \left( \sum_{\sigma=\pm 1} \sigma p_{\sigma} \left[ \frac{(C_{\sigma} \alpha_l)_{nm}}{\sqrt{2\pi \tilde{\alpha}_{l,\sigma}}} \exp \left[ -\frac{1}{2} \left( \frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}} \right)^2 \right] \right] \right)

- \sum_{\sigma=\pm 1} \sigma p_{\sigma} \left[ \phi \left( \frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}} \right) \right] Q_{lm}

+ \sum_{\sigma=\pm 1} p_{\sigma} \left[ \frac{(C_{\sigma} \alpha_m)_{nm}}{\sqrt{2\pi \tilde{\alpha}_{m,\sigma}}} \exp \left[ -\frac{1}{2} \left( \frac{\tilde{\beta}_{m,\sigma}}{\tilde{\alpha}_{m,\sigma}} \right)^2 \right] \right]

- \sum_{\sigma=\pm 1} p_{\sigma} \left[ \phi \left( \frac{\tilde{\beta}_{m,\sigma}}{\tilde{\alpha}_{m,\sigma}} \right) Q_{lm} \right]

Where,

\[ n_{bm} = \begin{cases} 3 & \text{if } m = 1 \\ 4 & \text{if } m = -1 \end{cases} \]

\[ n_{hm} = \begin{cases} 1 & \text{if } m = 1 \\ 2 & \text{if } m = -1 \end{cases} \]

\[ \tilde{\alpha}_{l,\sigma} = \sqrt{\alpha_l^T C_{\sigma} \alpha_l}, \quad \tilde{\beta}_{l,\sigma} = \alpha_l \cdot \mu_{\sigma} - \beta_l \]

\[ \tilde{\alpha}_{m,\sigma} = \sqrt{\alpha_m^T C_{\sigma} \alpha_m}, \quad \tilde{\beta}_{m,\sigma} = \alpha_m \cdot \mu_{\sigma} - \beta_m \]

4 Further Aspects

4.1 Generalization Error

For evaluation of any classification algorithm one of the most important criteria used is the generalization error. In the following we show how the generalization
error can be expressed in terms of order parameters. We define generalization error as the sum of TYPE-I and TYPE-II statistical error. Mathematically we define it as follows,

$$
\varepsilon_g \equiv \sum_{k=\pm 1} p_{-k} < \Theta_k >_{-k}
$$

where, $\Theta_k = \Theta \left( (\xi - w_{-k})^2 - (\xi - w_k)^2 \right)$ and $p_k$ is the prior probabilities of the class $k$.

To compute the averages $< \Theta_k >_{-k}$ let us look at the $\Theta$-function in this case.

$$
\Theta_k = \Theta \left( (\xi - w_{-k})^2 - (\xi - w_k)^2 \right)
= \Theta \left( -2\xi \cdot w_{-k} + w_{-k} \cdot w_{-k} + 2\xi \cdot w_k - w_k \cdot w_k \right)
= \Theta \left( -2h_{-k} + Q_{-k,-k} + 2h_k - Q_{k,k} \right)
= \Theta \left( -2, 2, 0, 0 \right) \cdot (h_{-k}, h_k, b_{-k}, b_k) + Q_{-k,-k} - Q_{k,k}
$$

Hence we have,

$$
\Theta_k = \Theta \left( \alpha_k \cdot x - \beta_k \right)
$$

Where, $\alpha_k = (2k, -2k, 0, 0)$ and $\beta_k = (Q_{k,k} - Q_{-k,-k})$.

Defining, $\tilde{\alpha}_{k,-k} = \sqrt{\alpha_k C_{-k} \alpha_k^T}$ and $\tilde{\beta}_{k,-k} = \alpha_k \cdot \mu_{-k} - \beta_k$, using (65) in Appendix A we have,

$$
< \Theta_k >_{-k} = \Phi \left( \frac{\tilde{\beta}_{k,-k}}{\tilde{\alpha}_{k,-k}} \right)
$$

Hence,

$$
\varepsilon_g \equiv \sum_{k=\pm 1} p_{-k} \Phi \left( \frac{\tilde{\beta}_{k,-k}}{\tilde{\alpha}_{k,-k}} \right)
$$

Another interesting quantity for studying the evolution process of the prototypes along the learning steps is the Euclidean distance between the prototypes and class mean vectors, which can also expressed in terms of order parameters as follows:

$$
d_{l,k} = (w_l - \lambda B_k)^2
= (w_l \cdot w_l - 2\lambda w_l \cdot B_k + \lambda^2 B_k \cdot B_k)
= (Q_{l,l} - 2\lambda R_{l,k} + \lambda^2) \cdot B_k \cdot B_k = 1
$$
4.2 Ineffectiveness of the window rule for LVQ2.1 algorithm for $N \to \infty$

If the prior probabilities are skewed the prototype vector corresponding to the class with lower probability gets push far away in the course of training process in the case of LVQ2.1. In practice this divergence problem is tackled by using a window. The prototypes are updated iff the data point $\xi^\mu$ at time stamp $\mu$ if it falls into a window, i.e the following holds [2]:

$$
\min \left( \frac{d(\xi^\mu, w_\sigma)}{d(\xi^\mu, w_{-\sigma})}, \frac{d(\xi^\mu, w_{-\sigma})}{d(\xi^\mu, w_\sigma)} \right) > s, s = \frac{1 - \gamma}{1 + \gamma}, 0 < \gamma \leq 1 \quad (47)
$$

where, $d$ is the Euclidean distance metric. Since,

$$
\lim_{N \to \infty} \min \left( \frac{d(\xi^\mu, w_\sigma)}{d(\xi^\mu, w_{-\sigma})}, \frac{d(\xi^\mu, w_{-\sigma})}{d(\xi^\mu, w_\sigma)} \right) = 1 \quad (48)
$$

for very high dimensionality of the data the above mentioned window does not work.

4.3 Best linear decision boundary and corresponding generalization error

Computing the Bayes optimal error for unequal class variances is a hard problem since the optimal decision surface is in general quadratic. However for a two class problem the generalization error can be approximated using Chernoff bound or Bhattacharya bound [4].

For a two class problem the decision boundary given by any LVQ algorithm is always linear or piece wise linear if we consider more than one prototype per class. In the model we use in this study the decision surface is always a linear one. Hence it is more logical to compare the performance of the algorithms (in terms of generalization error) with that of a best possible linear decision surface. To compute the generalization error for the best linear decision surface we consider the configuration depicted in Fig. 1 and parameterize the prototype vectors in the following way:

$$
w_1 = (d + g)B_1 \text{ and } w_{-1} = (d - g)B_{-1}.
$$

The above parameterization leads to parameterization of $\{R_{lm}, Q_{lm}\}$. Hence the generalization for the configuration shown in Fig. 1 can be computed to be the following:

$$
\varepsilon_g = p_1 \Phi \left( \frac{g - \lambda}{\sqrt{\nu_1}} \right) + p_{-1} \Phi \left( \frac{-g - \lambda}{\sqrt{\nu_{-1}}} \right) \quad (49)
$$

And the generalization corresponding to the best linear decision surface is calculated as follows:

$$
\varepsilon_{g,bld} = \min \{ \varepsilon_g | g \in \mathbb{R} \} \quad (50)
$$
Fig. 1. Best linear decision boundary and corresponding generalization error.

We should note that though in the configuration illustrated Fig. 1 we have taken $B_1 = -B_{-1}$ yet the generalization error is independent of the directions of the mean vectors provided we properly scale the magnitude ($\lambda$) of them i.e. we keep the distance between class centers constant. We also emphasize that $\tilde{\varepsilon}_{g,bld}$ matches with the Bayes optimal generalization error for $v_1 = v_{-1}$.

5 Summary and conclusions

For a simple and well behaved probability density function of the training data the dynamics of very high dimensional learning vector quantization (LVQ) algorithms can be theoretically analyzed using a few order parameters. Under the Markovian assumption these order parameters becomes self-averaging ([1]) and descriptions in terms of mean values are sufficient. The crucial step of such an analysis is the construction of solvable (analytically or numerically) system of coupled differential equations in terms of the order parameters, solving which we get an analytical description of the dynamics of the system. In this report we presented a detailed derivation of such differential equations for three different LVQ algorithms. The analysis is presented in a generic way, so that it can be helpful to use the same derivation and analysis for other types of learning algorithms.
References

15. N. N. R. Centre Bibliography on the Self-Organizing Maps (SOM) and Learning Vector Quantization (LVQ) *Otaniemi: Helsinki University of Technology.* Available o-line: http://liinwww.ira.uka.de/bibliography/Neural/SOM.LVQ.html

\[ A \langle x \rangle_n \Theta_s \rangle_k \text{ and } \langle \Theta_s \rangle_k \]
\[ \langle x \rangle_n \Theta_s \rangle_k = \frac{1}{(2\pi)^{d/2}(\det(C_k))^{1/2}} \int_{\mathbb{R}^d} (x \rangle_n \Theta(\alpha_s \cdot x - \beta_s) \]
\[ \exp \left( -\frac{1}{2} (x - \mu_k)^T C_k^{-1} (x - \mu_k) \right) dx \]
\[ = \frac{1}{(2\pi)^{d/2}(\det(C_k))^{1/2}} \int_{\mathbb{R}^d} (x' + \mu_k) \rangle_n \Theta(\alpha_s \cdot x' + \alpha_s \cdot \mu_k - \beta_s) \]
\[ \exp \left( -\frac{1}{2} x' \cdot C_k^{-1} x' \right) dx' \]
\( x' = x - \mu_k \). Substitute, \( x' = C_k^{\frac{1}{2}} y \). Where \( C_k^{\frac{1}{2}} \) is defined in following way, \( C_k = C_k^{\frac{1}{2}} C_k^{\frac{1}{2}} \). Since \( C_k \) is a covariance matrix, it is positive semidefinite, hence \( C_k^{\frac{1}{2}} \) exists. Hence we have \( dx' = \det(C_k^{\frac{1}{2}}) dy = (\det(C_k))^{\frac{1}{2}} dy \). Hence (51) =,

\[
\begin{align*}
&= \frac{1}{(2\pi)^2} \int_{\mathbb{R}^4} (C_k^{\frac{1}{2}} y)_n \Theta(\alpha_s C_k^{\frac{1}{2}} y + \alpha_s \cdot \mu_k - \beta_s) \\
&\quad \exp \left( -\frac{1}{2} y^2 \right) dy + (\mu_k)_n < \Theta_k >_k \\
&= \frac{1}{(2\pi)^2} \int_{\mathbb{R}^4} \sum_{j=1}^4 \left( (C_k^{\frac{1}{2}})_n (y)_j \right) \Theta(\alpha_s C_k^{\frac{1}{2}} y + \alpha_s \cdot \mu_k - \beta_s) \\
&\quad \exp \left( -\frac{1}{2} y^2 \right) dy + (\mu_k)_n < \Theta_k >_k \\
&= I + (\mu_k)_n < \Theta_k >_k \\
&= \frac{1}{(2\pi)^2} \int_{\mathbb{R}^4} \sum_{j=1}^4 \left( (C_k^{\frac{1}{2}})_n (y)_j \right) \Theta(\alpha_s C_k^{\frac{1}{2}} y + \alpha_s \cdot \mu_k - \beta_s) \\
&\quad \exp \left( -\frac{1}{2} y^2 \right) dy \\
\end{align*}
\]

Where

\[
I = \frac{1}{(2\pi)^2} \int_{\mathbb{R}^4} \sum_{j=1}^4 \left( (C_k^{\frac{1}{2}})_n (y)_j \right) \Theta(\alpha_s C_k^{\frac{1}{2}} y + \alpha_s \cdot \mu_k - \beta_s) \\
\quad \exp \left( -\frac{1}{2} y^2 \right) dy
\]

(52)

Define \( I_j = \int_{\mathbb{R}^4} (C_k^{\frac{1}{2}})_n (y)_j \Theta(\alpha_s C_k^{\frac{1}{2}} y + \alpha_s \cdot \mu_k - \beta_s) \exp \left( -\frac{1}{2} (y)_j^2 \right) d(y)_j \).

To compute \( I_j \) we use integration by parts, \( \int uv = \int udv - \int vdu \).

Take, \( u = \Theta(\alpha_s C_k^{\frac{1}{2}} y + \alpha_s \cdot \mu_k - \beta_s) \)

\[
u = (C_k^{\frac{1}{2}})_n \exp \left( -\frac{1}{2} (y)_j^2 \right)
\]

Hence, \( du = \frac{\partial}{\partial (y)_j} \Theta(\alpha_s C_k^{\frac{1}{2}} y + \alpha_s \cdot \mu_k - \beta_s) d(y)_j \)

\[
v = (-)(C_k^{\frac{1}{2}})_n \exp \left( -\frac{1}{2} (y)_j^2 \right) d(y)_j
\]

Hence we have,

\[
I_j = (-) \int u dv
\]

\[
= (-) \left[ \Theta(\alpha_s C_k^{\frac{1}{2}} y + \alpha_s \cdot \mu_k - \beta_s) (C_k^{\frac{1}{2}})_n \exp \left( -\frac{1}{2} (y)_j^2 \right) \right]_{-\infty}^{\infty}
\]
\[- \left[ \left( \int_{\mathbb{R}} (C_k^\dagger)_{nj} \frac{\partial}{\partial (y)_j} \left( \Theta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \exp \left( - \frac{1}{2} (y_j^2) \right) d(y)_j \right] \right.
\]
\[= 0 + \left[ \int_{\mathbb{R}} (C_k^\dagger)_{nj} \frac{\partial}{\partial (y)_j} \left( \Theta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \exp \left( - \frac{1}{2} (y_j^2) \right) d(y)_j \right. \]
\[= \int_{\mathbb{R}} (C_k^\dagger)_{nj} \frac{\partial}{\partial (y)_j} \left( \Theta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \exp \left( - \frac{1}{2} (y_j^2) \right) d(y)_j \]
\[= \int_{\mathbb{R}} (C_k^\dagger)_{nj} \frac{\partial}{\partial (y)_j} \left( \Theta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \exp \left( - \frac{1}{2} (y_j^2) \right) d(y)_j \]
\[= \int_{\mathbb{R}} (C_k^\dagger)_{nj} \frac{\partial}{\partial (y)_j} \left( \Theta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \exp \left( - \frac{1}{2} (y_j^2) \right) d(y)_j \]
\[= \int_{\mathbb{R}} (C_k^\dagger)_{nj} \frac{\partial}{\partial (y)_j} \left( \Theta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \exp \left( - \frac{1}{2} (y_j^2) \right) d(y)_j \]
\[= \int_{\mathbb{R}} (C_k^\dagger)_{nj} \frac{\partial}{\partial (y)_j} \left( \Theta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \exp \left( - \frac{1}{2} (y_j^2) \right) d(y)_j \]
\[= \int_{\mathbb{R}} (C_k^\dagger)_{nj} \frac{\partial}{\partial (y)_j} \left( \Theta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \exp \left( - \frac{1}{2} (y_j^2) \right) d(y)_j \]
\[= \int_{\mathbb{R}} (C_k^\dagger)_{nj} \frac{\partial}{\partial (y)_j} \left( \Theta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \exp \left( - \frac{1}{2} (y_j^2) \right) d(y)_j \]
\[= \int_{\mathbb{R}} (C_k^\dagger)_{nj} \frac{\partial}{\partial (y)_j} \left( \Theta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \exp \left( - \frac{1}{2} (y_j^2) \right) d(y)_j \]
\[= \int_{\mathbb{R}} (C_k^\dagger)_{nj} \frac{\partial}{\partial (y)_j} \left( \Theta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \exp \left( - \frac{1}{2} (y_j^2) \right) d(y)_j \]
\[= \int_{\mathbb{R}} (C_k^\dagger)_{nj} \frac{\partial}{\partial (y)_j} \left( \Theta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \exp \left( - \frac{1}{2} (y_j^2) \right) d(y)_j \]
\[= \int_{\mathbb{R}} (C_k^\dagger)_{nj} \frac{\partial}{\partial (y)_j} \left( \Theta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \exp \left( - \frac{1}{2} (y_j^2) \right) d(y)_j \]

Hence,
\[I = \frac{1}{(2\pi)^2} \sum_{j=1}^{4} (C_k^\dagger)_{nj} \int_{\mathbb{R}^4} \frac{\partial}{\partial (y)_j} \left( \Theta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \exp \left( - \frac{1}{2} (y_j^2) \right) d(y). \]

Define,
\[\Theta_j = \frac{\partial}{\partial (y)_j} \left( \Theta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \]
\[= \sum_{i=1}^{4} (\alpha_s)_i (C_k^\dagger)_{i,j} \left( \delta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \]

Where $\delta(\cdot)$ is the Dirac-delta function. Hence,
\[I = \frac{1}{(2\pi)^2} \sum_{j=1}^{4} (C_k^\dagger)_{nj} \sum_{i=1}^{4} (\alpha_s)_i (C_k^\dagger)_{i,j} \int_{\mathbb{R}^4} \left( \delta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \exp \left( - \frac{1}{2} (y_j^2) \right) d(y). \]
\[= \frac{1}{(2\pi)^2} (C_k \alpha_s) \int_{\mathbb{R}^4} \left( \delta (\alpha_s C_k^\dagger y + \alpha_s \cdot \mu_k - \beta_s) \right) \]
\[ \exp \left( -\frac{1}{2}y^2 \right) dy \] (57)

\[ \exp \left[ -\frac{1}{2}y^2 \right] dy \text{ is a measure which is invariant under rotation of the coordinate system. We rotate the coordinate system in such a way that one of the axes, say } \tilde{y} \text{ is along the vector } C_k^\parallel \alpha_s. \text{ Since } \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \exp \left[ -\frac{1}{2}z^2 \right] dz = 1, \]

\[ I = \frac{1}{\sqrt{2\pi}} (C_k^\parallel \alpha_s)_n \int_{\mathbb{R}} \delta \left( \|C_k^\parallel \alpha_s\| \tilde{y} + \alpha_s \cdot \mu_k - \beta_s \right) \exp \left[ -\frac{1}{2}y^2 \right] dy \] (58)

\[ \|C_k^\parallel \alpha_s\| = \sqrt{\alpha_s C_k \alpha_s} = \tilde{\alpha}_{s,k}, \text{ (say) and } \alpha_s \cdot \mu_k - \beta_s = \tilde{\beta}_{s,k} \text{ (say), then} \]

\[ I = \frac{1}{\sqrt{2\pi}} (C_k^\parallel \alpha_s)_n \int_{\mathbb{R}} \delta (\tilde{\alpha}_{s,k} \tilde{y} + \tilde{\beta}_{s,k}) \exp \left[ -\frac{1}{2}(\tilde{y})^2 \right] d\tilde{y}, \text{Put } z = \tilde{\alpha}_{s,k} \tilde{y} \]

\[ = \frac{(C_k^\parallel \alpha_s)_n}{\sqrt{2\pi \tilde{\alpha}_{s,k}}} \int_{\mathbb{R}} \delta (z + \tilde{\beta}_{s,k}) \exp \left[ -\frac{1}{2}(\frac{z}{\tilde{\alpha}_{s,k}})^2 \right] dz \]

\[ = \frac{(C_k^\parallel \alpha_s)_n}{\sqrt{2\pi \tilde{\alpha}_{s,k}}} \exp \left[ -\frac{1}{2}(\tilde{\beta}_{s,k})^2 \right] \] (59)

Since \( \int_{\mathbb{R}} \delta (x - a) f(x) dx = f(a) \),

\[ < (x)_n \Theta_s >_k = \frac{(C_k^\parallel \alpha_s)_n}{\sqrt{2\pi \tilde{\alpha}_{s,k}}} \exp \left[ -\frac{1}{2}(\tilde{\beta}_{s,k})^2 \right] + (\mu_k)_n < \Theta_s >_k \] (60)

Where,

\[ \tilde{\alpha}_{s,k} = \sqrt{\alpha_s^T C_k \alpha_s} \] (61)

\[ \tilde{\beta}_{s,k} = \alpha_s \cdot \mu_k - \beta_s \] (62)

Next we compute the following required average:

\[ < \Theta_s >_k \]

\[ = \frac{1}{(2\pi)^{4/2} (det(C_k))^{1/2}} \int_{\mathbb{R}^4} \Theta (\alpha_s \cdot x - \beta_s) \]

\[ \exp \left( -\frac{1}{2} (x - \mu_k)^T C_k^{-1} (x - \mu_k) \right) dx \]

\[ = \frac{1}{(2\pi)^{4/2} (det(C_k))^{1/2}} \int_{\mathbb{R}^4} \Theta (\alpha_s \cdot x' + \alpha_s \cdot \mu_k - \beta_s) \]

\[ \exp \left( -\frac{1}{2} x'^T C_k^{-1} x' \right) dx' \]
\[
\frac{1}{(2\pi)^2} \int_{\mathbb{R}^4} \Theta(\alpha_s C_k^\tau y + \alpha_s \cdot \mu_k - \beta_s) \exp \left( -\frac{1}{2} y^2 \right) dy
\]

\[
= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \Theta \left( \|C_k^\tau \alpha_s\| \bar{y} + \alpha_s \cdot \mu_k - \beta_s \right) \exp \left(-\frac{1}{2} \bar{y}^2 \right) d\bar{y}
\]

\[
= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \Theta(\tilde{\alpha}_{s,k} \bar{y} + \tilde{\beta}_{s,k}) \exp \left[ -\frac{1}{2} \bar{y}^2 \right] d\bar{y}
\]

(63)

\[
\Theta(\tilde{\alpha}_{s,k} \bar{y} + \tilde{\beta}_{s,k}) = 1 \text{ if } \tilde{\alpha}_{s,k} \bar{y} + \tilde{\beta}_{s,k} > 0 \Rightarrow \bar{y} > -\frac{\tilde{\beta}_{s,k}}{\tilde{\alpha}_{s,k}} \text{ and } \Theta(\tilde{\alpha}_{s,k} \bar{y} + \tilde{\beta}_{s,k}) = 0, \text{ otherwise. Hence,}
\]

\[
= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp \left[ -\frac{1}{2} \bar{y}^2 \right] d\bar{y}
\]

\[
= 1 - \Phi \left( -\frac{\tilde{\beta}_{s,k}}{\tilde{\alpha}_{s,k}} \right)
\]

\[
= \Phi \left( \frac{\tilde{\beta}_{s,k}}{\tilde{\alpha}_{s,k}} \right)
\]

(64)

Where \( \Phi(.) \) is the standard normal \((N(0,1))\) (cumulative) distribution function. Hence finally we have the required averages as follows,

\[
< \Theta_s >_k = \Phi \left( \frac{\tilde{\beta}_{s,k}}{\tilde{\alpha}_{s,k}} \right)
\]

\[
< (x)_n \Theta_s >_k = \frac{(C_k \alpha_s)_n}{\sqrt{2\pi \tilde{\alpha}_{s,k}}} \exp \left[ -\frac{1}{2} \left( \frac{\tilde{\beta}_{s,k}}{\tilde{\alpha}_{s,k}} \right)^2 \right] + (\mu_k)_n \Phi \left( \frac{\tilde{\beta}_{s,k}}{\tilde{\alpha}_{s,k}} \right)
\]

(65)

Where,

\[
\tilde{\alpha}_{s,k} = \sqrt{\alpha_s^T C_k \alpha_s}
\]

\[
\tilde{\beta}_{s,k} = \alpha_s \cdot \mu_k - \beta_s
\]