A theoretical framework for analysing the dynamics of LVQ

Ghosh, A.; Biehl, M.; Freking, A.; Reents, G.

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 $\Omega = \{\boldsymbol{w_1}, \ldots, \boldsymbol{w_c}\} \subset \mathbb{R}^N, c \geq \tilde{c}$, be the set of prototypes, where each $\omega_i \in \Omega$ is labeled with one of the $\tilde{c}$ classes. The NPC assigns any unlabeled object $\boldsymbol{\xi} \in \mathbb{R}^N$ 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 Ω. 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 Ω which will "best" represent the classes. The number of classes($\tilde{c}$) and the number of prototypes, (the cardinality of Ω, c), are predefined. A labeled training dataset of the form $(\boldsymbol{\xi}^\mu, \sigma^\mu)$ (where $\boldsymbol{\xi}^\mu \in \mathbb{R}^N$ is the training data vector presented to the on-line learning

⋆ Technical Report: 2004-9-02; Institute of Mathematics and Computing Science, University of Groningen P.O.Box. 800, 9700 AV Groningen, The Netherlands. anarta@cs.rug.nl. Last updated on 23rd June 2006.

algorithm at time stamp μ and σ^μ is its class label) is used to learn the elements of Ω. 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:

$$\begin{aligned} \boldsymbol{w_l}^\mu &= \boldsymbol{w_l}^{\mu-1} + \Delta\boldsymbol{w_l}^\mu, \\ &= \boldsymbol{w_l}^{\mu-1} + \frac{\eta}{N} f(\{\boldsymbol{w_l}^{\mu-1}\}, \boldsymbol{\xi}^\mu, \sigma^\mu)(\boldsymbol{\xi}^\mu - \boldsymbol{w_i}^\mu), l = 1 \dots c, \mu = 1, 2 \dots \end{aligned} \quad (1)$$

Where η 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 ± 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: $\boldsymbol{w}_s \in \mathbb{R}^N, s \in \{+1, -1\}$ - the prototype vectors for two different classes.

Data Model: $(\boldsymbol{\xi}^\mu, \sigma^\mu)$ is the model of the data. $\boldsymbol{\xi}^\mu$ is the input data vector. μ is the time stamp at which the data is observed, $\mu = 1, 2 \ldots$.
σ^μ is the class to which the observed data $\boldsymbol{\xi}^\mu$ belongs to at time step μ, $\sigma^\mu \in \{+1, -1\}$. We assume that $\boldsymbol{\xi}^\mu$ is independently drawn from a dataset having a probability density function $p(\boldsymbol{\xi})$ which is defined as follows:
$p(\boldsymbol{\xi}) = \sum_{\sigma=\pm 1} p_\sigma p(\boldsymbol{\xi}|\sigma)$, p_σ is the prior probabilities of the two classes, i.e. $p_{+1} = P(\sigma = +1)$ and $p_{-1} = P(\sigma = -1)$; $p(\boldsymbol{\xi}|\sigma)$ is the conditional probability density of the data given the class σ. In our study we choose $p(\boldsymbol{\xi}|\sigma)$ to be normally distributed with independent components (the covariance matrix is diagonal), i.e

$$p(\boldsymbol{\xi}|\sigma) = \frac{1}{(2\pi v_\sigma)^{\frac{N}{2}}} exp[-\frac{1}{2}\frac{(\boldsymbol{\xi} - \lambda \boldsymbol{B_\sigma})^2}{v_\sigma}], \tag{2}$$

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

$$< . >= \sum_{\sigma=\pm 1} p_\sigma < . >_\sigma \tag{3}$$

Where $< . >_\sigma$ is the conditional average for class σ. We choose the input data, $\boldsymbol{\xi}$, in such a way that,

$$\begin{aligned} < \boldsymbol{\xi} \cdot \boldsymbol{\xi} > &= \sum_{\sigma=\pm 1} p_\sigma < \boldsymbol{\xi}^2 >_\sigma \\ &= p_1(Nv_1 + \lambda^2) + p_{-1}(Nv_{-1} + \lambda^2) \\ &\approx N(p_1 v_1 + p_{-1} v_{-1}) \\ &\quad (\because N >> \lambda, < \boldsymbol{\xi}^2 >_\sigma \approx N v_\sigma) \end{aligned} \tag{4}$$

2.2 Order Parameters and Projections

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

$$\begin{aligned} R_{lm} &= \boldsymbol{w}_l \cdot \boldsymbol{B}_m \\ Q_{lm} &= \boldsymbol{w}_l \cdot \boldsymbol{w}_m \end{aligned} \tag{5}$$

$$\begin{aligned} b_l &= \boldsymbol{\xi} \cdot \boldsymbol{B}_l \\ h_l &= \boldsymbol{\xi} \cdot \boldsymbol{w}_l \end{aligned} \tag{6}$$

Define,

$$\boldsymbol{x} = (h_1, h_{-1}, b_1, b_{-1}) \tag{7}$$

Instantaneous projections: $b_l^\mu = \boldsymbol{\xi}^\mu \cdot \boldsymbol{B}_l$, $h_l^\mu = \boldsymbol{w}_l^{\mu-1} \cdot \boldsymbol{\xi}^\mu$
Instantaneous order parameters: $R_{lm}^\mu = \boldsymbol{w}_l^\mu \cdot \boldsymbol{B}_m$, $Q_{lm}^\mu = \boldsymbol{w}_l^\mu \cdot \boldsymbol{w}_m^\mu$

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, $p(h_{+1}, h_{-1}, b_{+1}, b_{-1}) = p(\boldsymbol{x})$, is normally distributed and fully specified by the corresponding conditional averages and covariances.

First Order Statistics of $\boldsymbol{h}$:

$$\begin{aligned} < h_l >_k &= \int_{\mathbb{R}^N} \boldsymbol{\xi} \cdot \boldsymbol{w}_l p(\boldsymbol{\xi}|\sigma = k) d\boldsymbol{\xi} = \boldsymbol{w}_l \cdot \int_{\mathbb{R}^N} \boldsymbol{\xi} p(\boldsymbol{\xi}|\sigma = k) d\boldsymbol{\xi} \\ &= \boldsymbol{w}_l \cdot \lambda \boldsymbol{B}_k. = \lambda R_{lk} \end{aligned} \tag{8}$$

First Order Statistics of $\boldsymbol{b}$:

$$\begin{aligned} < b_l >_k &= \int_{\mathbb{R}^N} \boldsymbol{\xi} \cdot \boldsymbol{B}_l(\boldsymbol{\xi}|\sigma = k) d\boldsymbol{\xi} = \boldsymbol{B}_l \cdot \int_{\mathbb{R}^N} \boldsymbol{\xi} p(\boldsymbol{\xi}|\sigma = k) d\boldsymbol{\xi} \\ &= \boldsymbol{B}_l \cdot \lambda \boldsymbol{B}_k = \lambda T_{lk} (say) \end{aligned} \tag{9}$$

Where $T_{lk} \equiv \boldsymbol{B}_l \cdot \boldsymbol{B}_k$ is called the overlap parameter.

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

$$\boldsymbol{\mu}_{k=+1} = \begin{pmatrix} \lambda R_{1,1} \\ \lambda R_{-1,1} \\ \lambda T_{1,1} \\ \lambda T_{-1,1} \end{pmatrix} \text{ and } \boldsymbol{\mu}_{k=-1} = \begin{pmatrix} \lambda R_{1,-1} \\ \lambda R_{-1,-1} \\ \lambda T_{1,-1} \\ \lambda T_{-1,-1} \end{pmatrix} \tag{10}$$

Second Order Statistics of $\boldsymbol{h}$: $< h_l h_m >_k - < h_l >_k < h_m >_k$
To compute the conditional variance let us first look at the average,

$$\begin{aligned} < h_l h_m >_k &= < (\boldsymbol{w}_l \cdot \boldsymbol{\xi})(\boldsymbol{w}_m \cdot \boldsymbol{\xi}) >_k \\ &= \left\langle \left(\sum_{i=1}^{N} (\boldsymbol{w}_l)_i (\boldsymbol{\xi})_i \right) \left(\sum_{j=1}^{N} (\boldsymbol{w}_m)_j (\boldsymbol{\xi})_j \right) \right\rangle_k \\ &= \left\langle \sum_{i=1}^{N} (\boldsymbol{w}_l)_i (\boldsymbol{w}_m)_i (\boldsymbol{\xi})_i (\boldsymbol{\xi})_i + \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} (\boldsymbol{w}_l)_i (\boldsymbol{w}_m)_j (\boldsymbol{\xi})_i (\boldsymbol{\xi})_j \right\rangle_k \\ &= \sum_{i=1}^{N} (\boldsymbol{w}_l)_i (\boldsymbol{w}_m)_i < (\boldsymbol{\xi})_i (\boldsymbol{\xi})_i >_k + \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} (\boldsymbol{w}_l)_i (\boldsymbol{w}_m)_j < (\boldsymbol{\xi})_i (\boldsymbol{\xi})_j >_k \\ &= \sum_{i=1}^{N} (\boldsymbol{w}_l)_i (\boldsymbol{w}_m)_i \left[v_k + \lambda^2 (\boldsymbol{B}_k)_i (\boldsymbol{B}_k)_i \right] \end{aligned}$$

$$
\begin{aligned}
&+\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}(\boldsymbol{w}_l)_i(\boldsymbol{w}_m)_j\lambda^2(\boldsymbol{B}_k)_i(\boldsymbol{B}_k)_j \\
&\Bigg[\because \text{components of } \boldsymbol{\xi} \text{ have variance } v_k, \\
&\Rightarrow < (\boldsymbol{\xi})_i,(\boldsymbol{\xi})_i >_k - < (\boldsymbol{\xi})_i >_k < (\boldsymbol{\xi})_i >_k = v_k, \forall i \in \{1\dots N\} \\
&\Rightarrow < (\boldsymbol{\xi})_i,(\boldsymbol{\xi})_i >_k = v_k + < (\boldsymbol{\xi})_i >_k < (\boldsymbol{\xi})_i >_k \\
&\text{and they are independent} \\
&\Rightarrow < (\boldsymbol{\xi})_i,(\boldsymbol{\xi})_j >_k - < (\boldsymbol{\xi})_i >_k < (\boldsymbol{\xi})_j >_k = 0, \forall i,j \in \{1\dots N\}, i \neq j. \\
&\Rightarrow < (\boldsymbol{\xi})_i,(\boldsymbol{\xi})_j >_k = < (\boldsymbol{\xi})_i >_k < (\boldsymbol{\xi})_j >_k \Bigg] \\
&= v_k\sum_{i=1}^{N}(\boldsymbol{w}_l)_i(\boldsymbol{w}_m)_i + \lambda^2\sum_{i=1}^{N}(\boldsymbol{w}_l)_i(\boldsymbol{w}_m)_i(\boldsymbol{B}_k)_i(\boldsymbol{B}_k)_i \\
&\quad +\lambda^2\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}(\boldsymbol{w}_l)_i(\boldsymbol{w}_m)_j(\boldsymbol{B}_k)_i(\boldsymbol{B}_k)_j \\
&= v_k\boldsymbol{w}_l\cdot\boldsymbol{w}_m + \lambda^2(\boldsymbol{w}_l\cdot\boldsymbol{B}_k)(\boldsymbol{w}_m\cdot\boldsymbol{B}_k) \\
&= v_kQ_{lm} + \lambda^2R_{lk}R_{mk} \qquad (11)
\end{aligned}
$$

Hence we have,

$$
\begin{aligned}
< h_l,h_m >_k - < h_l >_k < h_m >_k &= v_kQ_{lm} + \lambda^2R_{lk}R_{mk} - \lambda^2R_{lk}R_{mk} \\
&= v_kQ_{lm} \qquad (12)
\end{aligned}
$$

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

$$
< b_lb_m >_k - < b_l >_k < b_m >_k = v_kT_{lm} + \lambda^2T_{lk}T_{mk} - \lambda^2T_{lk}T_{mk} = v_kT_{lm} \qquad (13)
$$

Covariance: $< h_lb_m >_k - < h_l >_k < b_m >_k$
To compute the conditional variance let us first look at the average,

$$
\begin{aligned}
< h_lb_m >_k &= < (\boldsymbol{w}_l\cdot\boldsymbol{\xi})(\boldsymbol{B}_m\cdot\boldsymbol{\xi}) >_k \\
&= v_k\boldsymbol{w}_l\cdot\boldsymbol{B}_m + \lambda^2(\boldsymbol{w}_l\cdot\boldsymbol{B}_k)(\boldsymbol{B}_m\cdot\boldsymbol{B}_k) \\
&= v_kR_{lm} + \lambda^2R_{lk}T_{mk}
\end{aligned} \qquad (14)
$$

Hence we have,

$$
\begin{aligned}
< h_l,b_m >_k - < h_l >_k < b_m >_k &= v_kR_{lm} + \lambda^2R_{lk}T_{mk} - \lambda^2R_{lk}T_{mk} \\
&= v_kR_{lm} \qquad (15)
\end{aligned}
$$

The conditional density of $\boldsymbol{x}$ for class k is $N(\boldsymbol{\mu}_k, C_k)$, where, $\boldsymbol{\mu}_k$ 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 $\boldsymbol{x}$ is a 4-dimensional Gaussian. The covariance matrix by C_k and can be explicitly expressed as follows:

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

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 $(\boldsymbol{\xi}^\mu, \sigma^\mu)$, 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 σ^μ, 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:

$$\begin{aligned} \boldsymbol{w}_l{}^\mu &= \boldsymbol{w}_l{}^{\mu-1} + (\Delta \boldsymbol{w}_l{}^\mu)_{lvq2.1} \\ &= \boldsymbol{w}_l{}^{\mu-1} + \frac{\eta}{N}(l\sigma^\mu)(\boldsymbol{\xi}^\mu - \boldsymbol{w}_l{}^{\mu-1}) \end{aligned} \tag{17}$$

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 $(\boldsymbol{\xi}^\mu, \sigma^\mu)$, this LVQ algorithm determines the nearest prototype, say $\boldsymbol{w}_l$ and the nearest prototype, $\boldsymbol{w}_{\sigma^\mu}$ which has the same class label as the data class label, σ^μ 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:

$$\begin{aligned} \boldsymbol{w}_l{}^\mu &= \boldsymbol{w}_l{}^{\mu-1} + (\Delta \boldsymbol{w}_l{}^\mu)_{RSLVQ} \\ &= \boldsymbol{w}_l{}^{\mu-1} + \frac{\eta}{N}(l\sigma^\mu)(\boldsymbol{\xi}^\mu - \boldsymbol{w}_l{}^{\mu-1})\Theta\big(d_{+\sigma^\mu} - d_{-\sigma^\mu}\big) \end{aligned} \tag{18}$$

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

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

$$d_{\sigma^\mu} = \big(\boldsymbol{\xi}^\mu - \boldsymbol{w}_{\sigma^\mu}^{\mu-1}\big)^2 \tag{19}$$

We denote $\Theta\big(d_{+\sigma^\mu} - d_{-\sigma^\mu}\big)$ by Θ_{σ^μ} in short.

Winner takes all algorithms In this LVQ algorithm for every training data point, $(\boldsymbol{\xi}^{\mu}, \sigma^{\mu})$, only (and always) the nearest prototype is updated according to the following prescription :

$$\boldsymbol{w}_l{}^{\mu} = \boldsymbol{w}_l{}^{\mu-1} + \frac{\eta}{N}[a + bl\sigma^{\mu}]\Theta\big(d_{-l} - d_l\big)(\boldsymbol{\xi}^{\mu} - \boldsymbol{w}_l{}^{\mu-1}) \tag{20}$$

where $l \in \{+1, -1\}, a, b \in \mathbb{R}$. We denote $\Theta\big(d_{-l} - d_l\big)$ by Θ_l in short. The afore mentioned *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 $[a + bl\sigma^{\mu}]$. 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, σ^{μ} does not play any role in the learning process and the winner is always moved towards the data point, $\boldsymbol{\xi}^{\mu}$. 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,

$$E^{\mu} = \frac{1}{2}\sum_{l=\pm 1}\Theta\big(d^{\mu}_{-l} - d_l{}^{\mu}\big)(\boldsymbol{\xi}^{\mu} - \boldsymbol{w}_l{}^{\mu-1})^2 \tag{21}$$

(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 ($N \to \infty$) 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 coupled differential equations. In the following subsections we present a detailed derivation of such system of coupled differential equation for the above mentioned (17, 18, 20) three different LVQ algorithms.

3.1 LVQ2.1

Recurrence relations for order parameters R:

$$\begin{aligned} R_{lm}^{\mu} &= \boldsymbol{w}_l^{\mu} \cdot \boldsymbol{B}_m \\ &= \left(\boldsymbol{w}_l^{\mu-1} + (\Delta \boldsymbol{w}_l{}^{\mu})_{lvq2.1} \right) \cdot \boldsymbol{B}_m \\ &= \boldsymbol{w}_l^{\mu-1} \cdot \boldsymbol{B}_m + \left((\Delta \boldsymbol{w}_l{}^{\mu})_{lvq2.1} \right) \cdot \boldsymbol{B}_m \\ &= R_{lm}^{\mu-1} + \frac{\eta}{N}(l\sigma^{\mu}) \left(b_m^{\mu} - R_{lm}^{\mu-1} \right) \end{aligned} \tag{22}$$

Recurrence relation for order Parameter Q:

$$\begin{aligned} Q_{lm}^{\mu} &= \boldsymbol{w}_l^{\mu} \cdot \boldsymbol{w}_m^{\mu} \\ &= \left(\boldsymbol{w}_l{}^{\mu-1} + (\Delta \boldsymbol{w}_l{}^{\mu})_{lvq2.1} \right) \cdot \left(\boldsymbol{w}_m{}^{\mu-1} + (\Delta \boldsymbol{w}_m{}^{\mu})_{lvq2.1} \right) \\ &= Q_{lm}^{\mu-1} + \boldsymbol{w}_l{}^{\mu-1} \cdot (\Delta \boldsymbol{w}_m{}^{\mu})_{lvq2.1} + (\Delta \boldsymbol{w}_l{}^{\mu})_{lvq2.1} \cdot \boldsymbol{w}_m{}^{\mu-1} \\ &+ (\Delta \boldsymbol{w}_l{}^{\mu})_{lvq2.1} \cdot (\Delta \boldsymbol{w}_m{}^{\mu})_{lvq2.1} \\ &= Q_{lm}^{\mu-1} + \frac{\eta}{N} \Bigg(l\sigma^{\mu} h_m^{\mu-1} - l\sigma^{\mu} Q_{lm}^{\mu-1} + m\sigma^{\mu} h_l^{\mu-1} - m\sigma^{\mu} Q_{lm}^{\mu-1} \\ &+ \frac{\eta}{N}(lm)\boldsymbol{\xi} \cdot \boldsymbol{\xi} \Bigg) \end{aligned} \tag{23}$$

Since the analysis is for very large N we neglect the terms of $O(\frac{1}{N^2})$ in (23).

Differential equations In the thermodynamic limit, i.e. $N \to \infty$, 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,

$$\frac{dR_{lm}}{d\alpha} = \eta(l)\bigg(< \sigma b_m > - < \sigma > R_{lm} \bigg) \tag{24}$$

Similarly to (24) for Q we have,

$$\begin{aligned} \frac{dQ_{lm}}{d\alpha} = \eta\Bigg(& l < \sigma h_m > - l < \sigma > Q_{lm} + m < \sigma h_l > \\ & - m < \sigma > Q_{lm} + \eta(lm)(p_1 v_1 + p_{-1} v_{-1}) \Bigg); \\ & \because < \boldsymbol{\xi} \cdot \boldsymbol{\xi} > \approx N(p_1 v_1 + p_{-1} v_{-1}) \end{aligned} \tag{25}$$

From (24) and (25) we see that we need to calculate the following averages :
A. $< \sigma b_m >$
$= \sum_{\sigma=\pm 1} \int_{\mathbb{R}^N} \sigma b_m p_\sigma P\big(\boldsymbol{\xi}|\sigma\big) d\boldsymbol{\xi}$
$= p_{+1} \int_{\mathbb{R}^N} b_m P\big(\boldsymbol{\xi}| + 1\big) d\boldsymbol{\xi} - p_{-1} \int_{\mathbb{R}^N} b_m P\big(\boldsymbol{\xi}| - 1\big) d\boldsymbol{\xi}$
$= \sum_{\sigma=\pm 1} \sigma p_\sigma \lambda T_{m,\sigma}$
B. $< \sigma h_m >$
$= \sum_{\sigma=\pm 1} \int_{\mathbb{R}^N} \sigma h_m p_\sigma P\big(\boldsymbol{\xi}|\sigma\big) d\boldsymbol{\xi}$
$= p_{+1} \int_{\mathbb{R}^N} h_m P\big(\boldsymbol{\xi}| + 1\big) d\boldsymbol{\xi} - p_{-1} \int_{\mathbb{R}^N} h_m P\big(\boldsymbol{\xi}| - 1\big) d\boldsymbol{\xi}$
$= \sum_{\sigma=\pm 1} \sigma p_\sigma \lambda R_{m,\sigma}$
C. $< \sigma >$
$= \sum_{\sigma=\pm 1} \int_{\mathbb{R}^N} \sigma p_\sigma P\big(\boldsymbol{\xi}|\sigma\big) d\boldsymbol{\xi}$
$= \sum_{\sigma=\pm 1} \sigma p_\sigma$

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.

$$\frac{dR_{lm}}{d\alpha} = \eta(l)\Bigg(\sum_{\sigma=\pm 1} \sigma p_\sigma \lambda T_{m,\sigma} - \sum_{\sigma=\pm 1} \sigma p_\sigma R_{lm} \Bigg) \tag{26}$$

$$\begin{aligned} \frac{dQ_{lm}}{d\alpha} = \eta\Bigg(& l \sum_{\sigma=\pm 1} \sigma p_\sigma \lambda R_{m,\sigma} - l \sum_{\sigma=\pm 1} \sigma p_\sigma Q_{lm} + m \sum_{\sigma=\pm 1} \sigma p_\sigma \lambda R_{l,\sigma} \\ & - m \sum_{\sigma=\pm 1} \sigma p_\sigma Q_{lm} + \eta(lm)(p_1 v_1 + p_{-1} v_{-1}) \Bigg) \end{aligned} \tag{27}$$

3.2 RSLVQ

Recurrence relation for order parameter R:

$$R_{lm}^{\mu} = \boldsymbol{w}_l^{\mu} \cdot \boldsymbol{B}_m$$

$$= \left(\boldsymbol{w}_l^{\mu-1} + (\Delta \boldsymbol{w}_l{}^{\mu})_{RSLVQ} \right) \cdot \boldsymbol{B}_m$$
$$= R_{lm}^{\mu-1} + \frac{\eta}{N}(l\sigma^{\mu})(b_m \Theta_{\sigma^{\mu}} - R_{lm}^{\mu-1}\Theta_{\sigma^{\mu}}) \tag{28}$$

Recurrence relation for order parameter Q:

$$\begin{aligned} Q_{lm}^{\mu} &= \boldsymbol{w}_l^{\mu} \cdot \boldsymbol{w}_m^{\mu} \\ &= \left(\boldsymbol{w}_l{}^{\mu-1} + (\Delta \boldsymbol{w}_l{}^{\mu})_{RSLVQ} \right) \cdot \left(\boldsymbol{w}_m{}^{\mu-1} + (\Delta \boldsymbol{w}_m{}^{\mu})_{RSLVQ} \right) \\ &= Q_{lm}^{\mu-1} + \frac{\eta}{N} \Bigg(l\sigma^{\mu} h_m^{\mu-1} - l\sigma^{\mu} Q_{lm}^{\mu-1} + m\sigma^{\mu} h_l^{\mu-1} - m\sigma^{\mu} Q_{lm}^{\mu-1} \\ &+ \eta(lm)\boldsymbol{\xi} \cdot \boldsymbol{\xi} \Bigg) \Theta_{\sigma^{\mu}} \end{aligned} \tag{29}$$

We neglect the terms of $O(\frac{1}{N^2})$ in (29).

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

$$\frac{dR_{lm}}{d\alpha} = \eta(l) \Big(< \sigma b_m \Theta_{\sigma} > - < \sigma \Theta_{\sigma} > R_{lm} \Big) \tag{30}$$

$$\begin{aligned} \frac{dQ_{lm}}{d\alpha} = \eta \Bigg(& l < \sigma h_m \Theta_{\sigma} > -l < \sigma \Theta_{\sigma} > Q_{lm} + m < \sigma h_l \Theta_{\sigma} > \\ & -m < \sigma \Theta_{\sigma} > Q_{lm} \\ & + (lm)\eta(p_1 v_1 < \Theta_{\sigma} >_1 + p_{-1} v_{-1} < \Theta_{\sigma} >_{-1}) \Bigg) \end{aligned} \tag{31}$$

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

$$\begin{aligned} \Theta_{\sigma} &= \Theta\big(d_{+\sigma} - d_{-\sigma}\big) \\ &= \Theta\Big(\big(\boldsymbol{\xi} - \boldsymbol{w}_{+\sigma}\big)^2 - \big(\boldsymbol{\xi} - \boldsymbol{w}_{-\sigma}\big)^2 \Big) \\ &= \Theta\Big(-2\boldsymbol{\xi} \cdot \boldsymbol{w}_{+\sigma} + \boldsymbol{w}_{+\sigma} \cdot \boldsymbol{w}_{+\sigma} + 2\boldsymbol{\xi} \cdot \boldsymbol{w}_{-\sigma} - \boldsymbol{w}_{-\sigma} \cdot \boldsymbol{w}_{-\sigma} \Big) \\ &= \Theta\Big(-2h_{+\sigma} + Q_{+\sigma,+\sigma} + 2h_{-\sigma} - Q_{-\sigma,-\sigma} \Big) \end{aligned}$$

$$= \Theta\Big(\big(-2, +2, 0, 0\big) \cdot \big(h_{+\sigma}, h_{-\sigma}, b_{+\sigma}, b_{-\sigma}\big) + Q_{+\sigma,+\sigma} - Q_{-\sigma,-\sigma} \Big) \tag{32}$$

Hence we have,

$$\Theta_\sigma = \Theta\big(\boldsymbol{\alpha}_\sigma \cdot \boldsymbol{x} - \beta_\sigma\big), \tag{33}$$

Where, $\boldsymbol{\alpha}_\sigma = \big(-2\sigma, 2\sigma, 0, 0\big)$ and $\beta_\sigma = -\big(Q_{\sigma,\sigma} - Q_{-\sigma,-\sigma}\big)$

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 : $< (\boldsymbol{x})_n \Theta_\sigma >_\sigma$ and $< \Theta_\sigma >_\sigma$, where, $(\boldsymbol{x})_n$ is the n^{th} component of the vector $\boldsymbol{x}$; $n \in \{1, 2, 3, 4\}$.

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

$$\begin{aligned}
\frac{dR_{lm}}{d\alpha} = \eta(l) \Bigg(& \sum_{\sigma=\pm 1} \sigma p_\sigma \Big[\frac{(C_\sigma \boldsymbol{\alpha}_\sigma)_{n_{bm}}}{\sqrt{2\pi}\tilde{\alpha}_{\sigma,\sigma}} \exp\big[-\frac{1}{2}(\frac{\tilde{\beta}_{\sigma,\sigma}}{\tilde{\alpha}_{\sigma,\sigma}})^2\big] \\
& + (\boldsymbol{\mu}_\sigma)_{n_{bm}} \Phi\big(\frac{\tilde{\beta}_{\sigma,\sigma}}{\tilde{\alpha}_{\sigma,\sigma}}\big) \Big] - \sum_{\sigma=\pm 1} \sigma p_\sigma \big[\Phi\big(\frac{\tilde{\beta}_{\sigma,\sigma}}{\tilde{\alpha}_{\sigma,\sigma}}\big] R_{lm} \Bigg)
\end{aligned} \tag{34}$$

$$\begin{aligned}
\frac{dQ_{lm}}{d\alpha} = \eta \Bigg(& l \sum_{\sigma=\pm 1} \sigma p_\sigma \Big[\frac{(C_\sigma \boldsymbol{\alpha}_\sigma)_{n_{hm}}}{\sqrt{2\pi}\tilde{\alpha}_{\sigma,\sigma}} \exp\big[-\frac{1}{2}(\frac{\tilde{\beta}_{\sigma,\sigma}}{\tilde{\alpha}_{\sigma,\sigma}})^2\big] + (\boldsymbol{\mu}_\sigma)_{n_{hm}} \Phi\big(\frac{\tilde{\beta}_{\sigma,\sigma}}{\tilde{\alpha}_{\sigma,\sigma}}\big) \Big] \\
& - l \sum_{\sigma=\pm 1} \sigma p_\sigma \big[\Phi\big(\frac{\tilde{\beta}_{\sigma,\sigma}}{\tilde{\alpha}_{\sigma,\sigma}}\big] Q_{lm} \\
& + m \sum_{\sigma=\pm 1} \sigma p_\sigma \Big[\frac{(C_\sigma \boldsymbol{\alpha}_\sigma)_{n_{hl}}}{\sqrt{2\pi}\tilde{\alpha}_{\sigma,\sigma}} \exp\big[-\frac{1}{2}(\frac{\tilde{\beta}_{\sigma,\sigma}}{\tilde{\alpha}_{\sigma,\sigma}})^2\big] + (\boldsymbol{\mu}_\sigma)_{n_{hl}} \Phi\big(\frac{\tilde{\beta}_{\sigma,\sigma}}{\tilde{\alpha}_{\sigma,\sigma}}\big) \Big] \\
& - m \sum_{\sigma=\pm 1} \sigma p_\sigma \big[\Phi\big(\frac{\tilde{\beta}_{\sigma,\sigma}}{\tilde{\alpha}_{\sigma,\sigma}}\big] Q_{lm} + (lm)\eta \sum_{\sigma=\pm 1} v_\sigma p_\sigma \Phi\big(\frac{\tilde{\beta}_{\sigma,\sigma}}{\tilde{\alpha}_{\sigma,\sigma}}\big) \Bigg)
\end{aligned} \tag{35}$$

Where,

$$n_{bm} = \begin{matrix} 3 \text{ if } m = 1 \\ 4 \text{ if } m = -1 \end{matrix}$$

$$n_{hm} = \begin{matrix} 1 \text{ if } m = 1 \\ 2 \text{ if } m = -1 \end{matrix}$$

$$\tilde{\alpha}_\sigma = \sqrt{\boldsymbol{\alpha}_\sigma{}^T C_\sigma \boldsymbol{\alpha}_\sigma},\ \tilde{\beta}_{\sigma,\sigma} = \boldsymbol{\alpha}_\sigma \cdot \boldsymbol{\mu}_\sigma - \beta_\sigma$$

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:

$$\frac{dR_{lm}}{d\alpha} = \eta \Big[a\Big(< b_m \Theta_l > - < \Theta_l > R_{lm} \Big)$$

$$+ bl\Big(< \sigma b_m \Theta_l > - < \sigma \Theta_l > R_{lm} \Big) \Bigg] \tag{36}$$

$$\begin{aligned}
\frac{dQ_{lm}}{d\alpha} = \eta \Bigg[& b\Big(l < \sigma h_m \Theta_l > - l < \sigma \Theta_l > Q_{lm} + m < \sigma h_l \Theta_m > \\
& - m < \sigma \Theta_m > Q_{lm} \Big) \\
& + a\Big(< h_m \Theta_l > - < \Theta_l > Q_{lm} + < h_l \Theta_m > - < \Theta_m > Q_{lm} \Big) \\
& + \delta_{lm} \eta \big[a^2 + b^2 lm \big] (p_1 v_1 < \Theta_l >_1 + p_{-1} v_{-1} < \Theta_l >_{-1}) \\
& + \delta_{lm} ab(l+m)(p_1 v_1 < \Theta_l >_1 + p_{-1} v_{-1} < \Theta_l >_{-1}) \Bigg]
\end{aligned} \tag{37}$$

Where,

$$\delta_{lm} = \begin{matrix} 1 \text{ if } m = l \\ 0 \text{ if } m \neq l \end{matrix}$$

Similar to Θ_σ of RSLVQ we can express Θ_l in the following way,

$$\Theta_l = \Theta\big(\boldsymbol{\alpha}_l \cdot \boldsymbol{x} - \beta_l \big) \tag{38}$$

Where $\boldsymbol{\alpha}_l = (2l, -2l, 0, 0)$ 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: $< (\boldsymbol{x})_n \Theta_l >_k$ and $< \Theta_l >_k$. Using averages in (65) in Appendix A we get the following expression for differential equations,

$$\begin{aligned}
\frac{dR_{lm}}{d\alpha} = \eta \Bigg[& (bl)\Big(\sum_{\sigma=\pm 1} \sigma p_\sigma \Big[\frac{(C_\sigma \boldsymbol{\alpha}_l)_{n_{bm}}}{\sqrt{2\pi}\tilde{\alpha}_{l,\sigma}} \exp\big[-\frac{1}{2} (\frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}})^2 \big] \\
& + (\boldsymbol{\mu}_\sigma)_{n_{bm}} \Phi(\frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}}) \Big] - \sum_{\sigma=\pm 1} \sigma p_\sigma \big[\Phi(\frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}} \big] R_{lm} \Big) \\
& + (a)\Big(\sum_{\sigma=\pm 1} p_\sigma \Big[\frac{(C_\sigma \boldsymbol{\alpha}_l)_{n_{bm}}}{\sqrt{2\pi}\tilde{\alpha}_{l,\sigma}} \exp\big[-\frac{1}{2} (\frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}})^2 \big] + (\boldsymbol{\mu}_\sigma)_{n_{bm}} \Phi(\frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}}) \Big] \\
& - \sum_{\sigma=\pm 1} p_\sigma \big[\Phi(\frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}} \big] R_{lm} \Big) \Bigg]
\end{aligned} \tag{39}$$

$$\begin{aligned}
\frac{dQ_{lm}}{d\alpha} = \eta \Big(& bl \sum_{\sigma=\pm 1} \sigma p_\sigma \Big[\frac{(C_\sigma \boldsymbol{\alpha}_l)_{n_{hm}}}{\sqrt{2\pi}\tilde{\alpha}_{l,\sigma}} \exp\big[-\frac{1}{2} (\frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}})^2 \big] \\
& + (\boldsymbol{\mu}_\sigma)_{n_{hm}} \Phi(\frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}}) \Big] \\
& - bl \sum_{\sigma=\pm 1} \sigma p_\sigma \big[\Phi(\frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}} \big] Q_{lm}
\end{aligned}$$

$$
\begin{aligned}
&+bm \sum_{\sigma=\pm 1} \sigma p_\sigma \Big[\frac{(C_\sigma \boldsymbol{\alpha}_l)_{n_{hl}}}{\sqrt{2\pi}\tilde{\alpha}_{m,\sigma}} \exp\big[-\frac{1}{2}(\frac{\tilde{\beta}_{m,\sigma}}{\tilde{\alpha}_{m,\sigma}})^2\big] \\
&+(\boldsymbol{\mu}_\sigma)_{n_{hl}} \Phi\big(\frac{\tilde{\beta}_{m,\sigma}}{\tilde{\alpha}_{m,\sigma}}\big)\Big] \\
&-bm \sum_{\sigma=\pm 1} \sigma p_\sigma \big[\Phi(\frac{\tilde{\beta}_{m,\sigma}}{\tilde{\alpha}_{m,\sigma}}\big] Q_{lm}\Big) \\
&+ \delta_{lm}(a^2 + b^2 lm)\eta^2 \sum_{\sigma=\pm 1} \sigma p_\sigma v_\sigma \Phi(\frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}}) \\
&+\delta_{lm}\eta^2\big(ab(l+m)\big) \sum_{\sigma=\pm 1} \sigma p_\sigma v_\sigma \Phi(\frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}}) \\
&+ \eta\Big(a \sum_{\sigma=\pm 1} p_\sigma \Big[\frac{(C_\sigma \boldsymbol{\alpha}_l)_{n_{hm}}}{\sqrt{2\pi}\tilde{\alpha}_{l,\sigma}} \exp\big[-\frac{1}{2}(\frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}})^2\big] \\
&+(\boldsymbol{\mu}_\sigma)_{n_{hm}} \Phi\big(\frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}}\big)\Big] \\
&-a \sum_{\sigma=\pm 1} p_\sigma \big[\Phi(\frac{\tilde{\beta}_{l,\sigma}}{\tilde{\alpha}_{l,\sigma}}\big] Q_{lm} \\
&+a \sum_{\sigma=\pm 1} p_\sigma \Big[\frac{(C_\sigma \boldsymbol{\alpha}_l)_{n_{hl}}}{\sqrt{2\pi}\tilde{\alpha}_{m,\sigma}} \exp\big[-\frac{1}{2}(\frac{\tilde{\beta}_{m,\sigma}}{\tilde{\alpha}_{m,\sigma}})^2\big] \\
&+(\boldsymbol{\mu}_\sigma)_{n_{hl}} \Phi\big(\frac{\tilde{\beta}_{m,\sigma}}{\tilde{\alpha}_{m,\sigma}}\big)\Big] \\
&-a \sum_{\sigma=\pm 1} p_\sigma \big[\Phi(\frac{\tilde{\beta}_{m,\sigma}}{\tilde{\alpha}_{m,\sigma}}\big] Q_{lm}\Big)
\end{aligned} \tag{40}
$$

Where,

$$
n_{bm} = \begin{matrix} 3 \text{ if } m = 1 \\ 4 \text{ if } m = -1 \end{matrix}
$$

$$
n_{hm} = \begin{matrix} 1 \text{ if } m = 1 \\ 2 \text{ if } m = -1 \end{matrix}
$$

$$
\begin{aligned}
\tilde{\alpha}_{l,\sigma} &= \sqrt{\boldsymbol{\alpha}_l{}^T C_\sigma \boldsymbol{\alpha}_l},\ \tilde{\beta}_{l,\sigma} = \boldsymbol{\alpha}_l \cdot \boldsymbol{\mu}_\sigma - \beta_l \\
\tilde{\alpha}_{m,\sigma} &= \sqrt{\boldsymbol{\alpha}_m{}^T C_\sigma \boldsymbol{\alpha}_m},\ \tilde{\beta}_{m,\sigma} = \boldsymbol{\alpha}_m \cdot \boldsymbol{\mu}_\sigma - \beta_m
\end{aligned}
$$

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,

$$\varepsilon_g \equiv \sum_{k=\pm 1} p_{-k} < \Theta_k >_{-k} \tag{41}$$

where, $\Theta_k = \Theta\Big(\big(\boldsymbol{\xi} - \boldsymbol{w}_{-k}\big)^2 - \big(\boldsymbol{\xi} - \boldsymbol{w}_k\big)^2\Big)$ and p_k is the prior probabilities of the class k.

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

$$\begin{aligned}
\Theta_k &= \Theta\Big(\big(\boldsymbol{\xi} - \boldsymbol{w}_{-k}\big)^2 - \big(\boldsymbol{\xi} - \boldsymbol{w}_k\big)^2\Big) \\
&= \Theta\Big(- 2\boldsymbol{\xi} \cdot \boldsymbol{w}_{-k} + \boldsymbol{w}_{-k} \cdot \boldsymbol{w}_{-k} + 2\boldsymbol{\xi} \cdot \boldsymbol{w}_k - \boldsymbol{w}_k \cdot \boldsymbol{w}_k\Big) \\
&= \Theta\Big(- 2h_{-k} + Q_{-k,-k} + 2h_k - Q_{k,k}\Big) \\
&= \Theta\Big(\big(- 2, 2, 0, 0\big) \cdot \big(h_{-k}, h_k, b_k, b_{-k}\big) + Q_{-k,-k} - Q_{k,k}\Big)
\end{aligned} \tag{42}$$

Hence we have,

$$\Theta_k = \Theta\Big(\alpha_k \cdot \boldsymbol{x} - \beta_k\Big) \tag{43}$$

Where, $\alpha_k = \Big(2k, -2k, 0, 0\Big)$ 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,

$$< \Theta_k >_{-k} = \Phi\Big(\frac{\tilde{\beta}_{k,-k}}{\tilde{\alpha}_{k,-k}}\Big) \tag{44}$$

Hence,

$$\varepsilon_g \equiv \sum_{k=\pm 1} p_{-k} \Phi\Big(\frac{\tilde{\beta}_{k,-k}}{\tilde{\alpha}_{k,-k}}\Big) \tag{45}$$

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:

$$\begin{aligned}
d_{l,k} &= \big(\boldsymbol{w}_l - \lambda \boldsymbol{B}_k\big)^2 \\
&= \big(\boldsymbol{w}_l \cdot \boldsymbol{w}_l - 2\lambda \boldsymbol{w}_l \cdot \boldsymbol{B}_k + \lambda^2 \boldsymbol{B}_k \cdot \boldsymbol{B}_k\big) \\
&= \big(Q_{l,l} - 2\lambda R_{l,k} + \lambda^2\big) \because \boldsymbol{B}_k \cdot \boldsymbol{B}_k = 1
\end{aligned} \tag{46}$$

4.2 Ineffectiveness of the window rule for LVQ2.1 algorithm for $N \rightarrow \infty$

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 $\boldsymbol{\xi}^{\mu}$ at time stamp μ if it falls into a window, i.e the following holds [2]:

$$min\Big(\frac{d(\boldsymbol{\xi}^{\mu}, \boldsymbol{w}_{\sigma})}{d(\boldsymbol{\xi}^{\mu}, \boldsymbol{w}_{-\sigma})}, \frac{d(\boldsymbol{\xi}^{\mu}, \boldsymbol{w}_{-\sigma})}{d(\boldsymbol{\xi}^{\mu}, \boldsymbol{w}_{\sigma})}\Big) > s, s = \frac{1-\gamma}{1+\gamma}, 0 < \gamma \leq 1 \tag{47}$$

where, d is the Euclidean distance metric. Since,

$$\lim_{N \rightarrow \infty} min\Big(\frac{d(\boldsymbol{\xi}^{\mu}, \boldsymbol{w}_{\sigma})}{d(\boldsymbol{\xi}^{\mu}, \boldsymbol{w}_{-\sigma})}, \frac{d(\boldsymbol{\xi}^{\mu}, \boldsymbol{w}_{-\sigma})}{d(\boldsymbol{\xi}^{\mu}, \boldsymbol{w}_{\sigma})}\Big) = 1 \tag{48}$$

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:

$$\boldsymbol{w}_1 = (d+g)\boldsymbol{B}_1 \text{ and } \boldsymbol{w}_{-1} = (d-g)\boldsymbol{B}_{-1}.$$

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:

$$\varepsilon_g = p_1 \Phi\big(\frac{g-\lambda}{\sqrt{v_1}}\big) + p_{-1}\Phi\big(\frac{-g-\lambda}{\sqrt{v_{-1}}}\big) \tag{49}$$

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

$$\varepsilon_{g,bld} = min\{\varepsilon_g | g \in \mathbb{R}\} \tag{50}$$

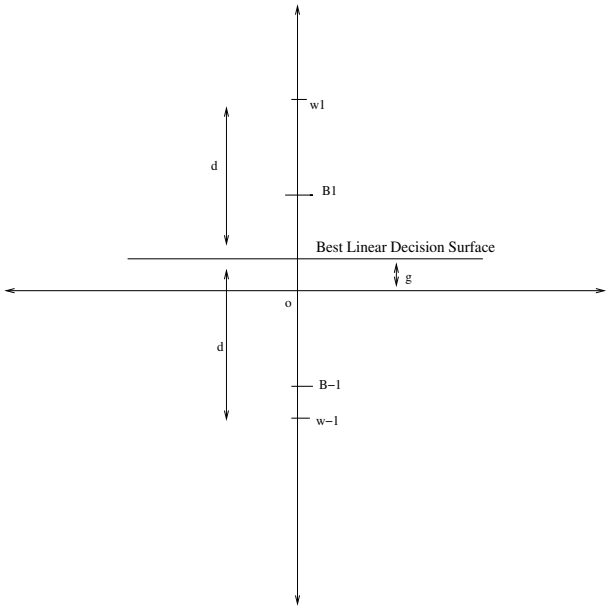

Fig. 1. Best linear decision boundary and corresponding generalization error.

We should note that though in the configuration illustrated Fig. 1 we have taken $\boldsymbol{B}_1 = -\boldsymbol{B}_{-1}$ yet the generalization error is independent of the directions of the mean vectors provided we properly scale the magnitude (λ) of them i.e. we keep the distance between class centers constant. We also emphasize that $\varepsilon_{g,bld}$ 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

1. G. Reents and R. Urbanczik. Self-Averaging and On-Line Learning *Physical review letters. Vol. 80. No. 24*, pp. 5445-5448, 1998.
2. S. Seo and K. Obermayer. Soft Learning Vector Quantization. *Neural Computing, 15*, pp. 1589-1604, 2003.
3. T. Kohonen. Learning Vector Quantization. *Technical Report, Otaniemi: Helsonki University of Technology.*
4. R. O. Duda and P. E. Hart Pattern Classification and Scene Analysis. *New York: Wiley* 2000.
5. T. Kohonen Learning Vector Quantization. *In M. Arbib, editor, The handbook of brain theory and neural networks. MIT Press.* pp. 537-540, 1995
6. B. Hammer and T. Villmann Generalized Relevance Learning Vector Quantization *Neural Network, Vol. 15.(8-9)* pp. 1059-1068, 2002
7. T. Kohonen Self-Organizing Maps *Springer, 1997*
8. A. S. Sato and K. Yamada Generalized Learning vector Quantization *In G. Tesauro, D. Touretzky and T. Leen, editors, Advances in Neural Information Processing Systems, Vol. 7* pp. 423-429, 1995
9. P. Somervuo and T. Kohonen Self-Organizing Maps and Learning Vector Quantization for Feature Sequences *Neural Processing Letters, 10(2)* pp. 151-159, 1999
10. V. R. de Sa and D. H. Ballard A Note on Learning Vector Quantization *In C. L. Giles, S. J. Hanson and J. D. Cowan, editors, Advances in Neural Information processing 5.*pp. 220-227, 1993
11. M. Opper and W. Kinzel Physiscs of Neural Network *edited by J. S. van Hemmen, E. Domany, and K. Schulten (Springer-Verlag, Berlin, to be published).*
12. T. H. L. Watkin A. Rau and M. Biehl The statistical mechanics of learning a rule *In Review of Modern Physiscs, 65, (1993) 499*
13. M. Biehl, A. Freking and G. Reents Dynamics of On-Line Competitive Learning *Europhysiscs Letters. Vol. 38. No. 1.* pp. 73-78, 1997
14. T. Kohonen Improved Versions of Learning Vector Quantization *IJCNN, International Joint conference on Neural Networks, Vol. 1* pp. 545-550, 1990
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 $< (x)_n \Theta_s >_k$ and $< \Theta_s >_k$

$< (x)_n \Theta_s >_k$

$$
\begin{aligned}
&= \frac{1}{(2\pi)^{4/2}\big(det(C_k)\big)^{\frac{1}{2}}} \int_{\mathbb{R}^4} (x)_n \Theta\big(\boldsymbol{\alpha}_s \cdot \boldsymbol{x} - \beta_s\big) \\
&\quad \exp\left(-\frac{1}{2}\big(\boldsymbol{x} - \boldsymbol{\mu}_k\big)^T C_k^{-1}\big(\boldsymbol{x} - \boldsymbol{\mu}_k\big)\right) d\boldsymbol{x} \\
&= \frac{1}{(2\pi)^{4/2}\big(det(C_k)\big)^{\frac{1}{2}}} \int_{\mathbb{R}^4} \big(x^{'} + \boldsymbol{\mu}_k\big)_n \Theta\big(\boldsymbol{\alpha}_s \cdot \boldsymbol{x}^{'} + \boldsymbol{\alpha}_s \cdot \boldsymbol{\mu}_k - \beta_s\big) \\
&\quad \exp\left(-\frac{1}{2}\boldsymbol{x}^{'T} C_k^{-1} \boldsymbol{x}^{'}\right) d\boldsymbol{x}^{'}
\end{aligned}
$$

(51)

$\boldsymbol{x}^{'} = \boldsymbol{x} - \boldsymbol{\mu}_k$. Substitute, $\boldsymbol{x}^{'} = {C_k}^{\frac{1}{2}}\boldsymbol{y}$. 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 $d\boldsymbol{x}^{'} = det(C_k^{\frac{1}{2}})d\boldsymbol{y} = (det(C_k))^{\frac{1}{2}}d\boldsymbol{y}$. Hence (51) =,

$$
\begin{aligned}
&= \frac{1}{(2\pi)^2}\int_{\mathbb{R}^4}(C_k^{\frac{1}{2}}\boldsymbol{y})_n\Theta\big(\boldsymbol{\alpha}_s C_k^{\frac{1}{2}}\boldsymbol{y} + \boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k - \beta_s\big) \\
&\quad \exp\left(-\frac{1}{2}\boldsymbol{y}^2\right)d\boldsymbol{y} + (\boldsymbol{\mu}_k)_n < \Theta_s >_k \\
&= \frac{1}{(2\pi)^2}\int_{\mathbb{R}^4}\sum_{j=1}^{4}\left((C_k^{\frac{1}{2}})_{nj}(\boldsymbol{y})_j\right)\Theta\big(\boldsymbol{\alpha}_s C_k^{\frac{1}{2}}\boldsymbol{y} + \boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k - \beta_s\big) \\
&\quad \exp\left(-\frac{1}{2}\boldsymbol{y}^2\right)d\boldsymbol{y} + (\boldsymbol{\mu}_k)_n < \Theta_s >_k \\
&= I + (\boldsymbol{\mu}_k)_n < \Theta_s >_k
\end{aligned}
\tag{52}
$$

Where

$$
\begin{aligned}
I = \frac{1}{(2\pi)^2}\int_{\mathbb{R}^4}\sum_{j=1}^{4}\left((C_k^{\frac{1}{2}})_{nj}(\boldsymbol{y})_j\right)\Theta\big(\boldsymbol{\alpha}_s C_k^{\frac{1}{2}}\boldsymbol{y} + \boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k - \beta_s\big) \\
\exp\left(-\frac{1}{2}\boldsymbol{y}^2\right)d\boldsymbol{y}
\end{aligned}
\tag{53}
$$

Define $I_j = \int_{\mathbb{R}}(C_k^{\frac{1}{2}})_{nj}(\boldsymbol{y})_j\Theta\big(\boldsymbol{\alpha}_s C_k^{\frac{1}{2}}\boldsymbol{y} + \boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k - \beta_s\big)\exp\left(-\frac{1}{2}(\boldsymbol{y})_j^2\right)d(\boldsymbol{y})_j$.
To compute I_j we use integration by parts, ($\int udv = uv - \int vdu$).
Take,
$u = \Theta\big(\boldsymbol{\alpha}_s C_k^{\frac{1}{2}}\boldsymbol{y} + \boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k - \beta_s\big)$
$v = (C_k^{\frac{1}{2}})_{nj}\exp\left(-\frac{1}{2}(\boldsymbol{y})_j^2\right)$
Hence,
$du = \frac{\partial}{\partial(\boldsymbol{y})_j}\Theta\big(\boldsymbol{\alpha}_s C_k^{\frac{1}{2}}\boldsymbol{y} + \boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k - \beta_s\big)d(\boldsymbol{y})_j$
$dv = (-)(C_k^{\frac{1}{2}})_{nj}(\boldsymbol{y})_j\exp\left(-\frac{1}{2}(\boldsymbol{y})_j^2\right)d(\boldsymbol{y})_j$
Hence we have,

$$
\begin{aligned}
I_j &= (-)\int udv \\
&= (-)\left[\Theta\big(\boldsymbol{\alpha}_s C_k^{\frac{1}{2}}\boldsymbol{y} + \boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k - \beta_s\big)(C_k^{\frac{1}{2}})_{nj}\exp\left(-\frac{1}{2}(\boldsymbol{y})_j^2\right)\right]_{-\infty}^{\infty}
\end{aligned}
$$

$$
\begin{aligned}
&-\Bigg[(-)\int_{\mathbb{R}}(C_k^{\frac{1}{2}})_{nj}\frac{\partial}{\partial(y)_j}\Bigg(\Theta\big(\boldsymbol{\alpha}_s C_k^{\frac{1}{2}}\boldsymbol{y}+\boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k-\beta_s\big)\Bigg)\\
&\exp\Bigg(-\frac{1}{2}(\boldsymbol{y})_j^2\Bigg)d(\boldsymbol{y})_j\Bigg]\\
&=0+\\
&\Bigg[\int_{\mathbb{R}}(C_k^{\frac{1}{2}})_{nj}\frac{\partial}{\partial(y)_j}\Bigg(\Theta\big(\boldsymbol{\alpha}_s C_k^{\frac{1}{2}}\boldsymbol{y}+\boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k-\beta_s\big)\Bigg)\\
&\exp\Bigg(-\frac{1}{2}(\boldsymbol{y})_j^2\Bigg)d(\boldsymbol{y})_j\Bigg]\\
&=\int_{\mathbb{R}}(C_k^{\frac{1}{2}})_{nj}\frac{\partial}{\partial(y)_j}\Bigg(\Theta\big(\boldsymbol{\alpha}_s C_k^{\frac{1}{2}}\boldsymbol{y}+\boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k-\beta_s\big)\Bigg)\\
&\exp\Bigg(-\frac{1}{2}(\boldsymbol{y})_j^2\Bigg)d(\boldsymbol{y})_j
\end{aligned}
\tag{54}
$$

Hence,

$$
\begin{aligned}
I=\frac{1}{(2\pi)^2}\sum_{j=1}^{4}(C_k^{\frac{1}{2}})_{nj}\int_{\mathbb{R}^4}\frac{\partial}{\partial(y)_j}\Bigg(\Theta\big(\boldsymbol{\alpha}_s C_k^{\frac{1}{2}}\boldsymbol{y}+\boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k-\beta_s\big)\Bigg)\\
\exp\Bigg(-\frac{1}{2}\boldsymbol{y}^2\Bigg)d\boldsymbol{y}.
\end{aligned}
\tag{55}
$$

Define,

$$
\begin{aligned}
\Theta_j^{'}&\equiv\frac{\partial}{\partial(y)_j}\Bigg(\Theta\big(\boldsymbol{\alpha}_s C_k^{\frac{1}{2}}\boldsymbol{y}+\boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k-\beta_s\big)\Bigg)\\
&=\sum_{i=1}^{4}(\boldsymbol{\alpha}_s)_i(C_k^{\frac{1}{2}})_{i,j}\Bigg(\delta\big(\boldsymbol{\alpha}_s C_k^{\frac{1}{2}}\boldsymbol{y}+\boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k-\beta_s\big)\Bigg)
\end{aligned}
\tag{56}
$$

Where $\delta(.)$ is the Dirac-delta function. Hence,

$$
\begin{aligned}
I&=\frac{1}{(2\pi)^2}\sum_{j=1}^{4}\Bigg((C_k^{\frac{1}{2}})_{nj}\sum_{i=1}^{4}(\boldsymbol{\alpha}_s)_i(C_k^{\frac{1}{2}})_{i,j}\Bigg)\int_{\mathbb{R}^4}\Bigg(\delta\big(\boldsymbol{\alpha}_s C_k^{\frac{1}{2}}\boldsymbol{y}+\boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k-\beta_s\big)\Bigg)\\
&\exp\Bigg(-\frac{1}{2}\boldsymbol{y}^2\Bigg)d\boldsymbol{y}.\\
&=\frac{1}{(2\pi)^2}(C_k\boldsymbol{\alpha}_s)_n\int_{\mathbb{R}^4}\Bigg(\delta\big(\boldsymbol{\alpha}_s C_k^{\frac{1}{2}}\boldsymbol{y}+\boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k-\beta_s\big)\Bigg)
\end{aligned}
$$

$$\exp\left(-\frac{1}{2}\boldsymbol{y}^2\right)d\boldsymbol{y} \tag{57}$$

$\exp\big[-\frac{1}{2}\boldsymbol{y}^2\big]d\boldsymbol{y}$ 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 $\tilde{y}$ is along the vector $C_k^{\frac{1}{2}}\boldsymbol{\alpha}_s$. Since $\frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}}\exp\big[-\frac{1}{2}z^2\big]dz = 1$,

$$I = \frac{1}{\sqrt{2\pi}}(C_k\boldsymbol{\alpha}_s)_n\int_{\mathbb{R}}\delta\left(\|C_k^{\frac{1}{2}}\boldsymbol{\alpha}_s\|\tilde{y} + \boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k - \beta_s\right)\exp\big[-\frac{1}{2}\tilde{y}^2\big]d\tilde{y} \tag{58}$$

$\|C_k^{\frac{1}{2}}\boldsymbol{\alpha}_s\| = \sqrt{\boldsymbol{\alpha}_s C_k\boldsymbol{\alpha}_s} = \tilde{\alpha}_{s,k}$, (say) and $\boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k - \beta_s = \tilde{\beta}_{s,k}$ (say), then

$$\begin{aligned}
I &= \frac{1}{\sqrt{2\pi}}(C_k\boldsymbol{\alpha}_s)_n\int_{\mathbb{R}}\delta\big(\tilde{\alpha}_{s,k}\tilde{y} + \tilde{\beta}_{s,k}\big)\exp\big[-\frac{1}{2}\tilde{y}^2\big]d\tilde{y} \text{ ,Put } z = \tilde{\alpha}_{s,k}\tilde{y}\\
&= \frac{(C_k\boldsymbol{\alpha}_s)_n}{\sqrt{2\pi}\tilde{\alpha}_{s,k}}\int_{\mathbb{R}}\delta\big(z + \tilde{\beta}_{s,k}\big)\exp\big[-\frac{1}{2}(\frac{z}{\tilde{\alpha}_{s,k}})^2\big]dz\\
&= \frac{(C_k\boldsymbol{\alpha}_s)_n}{\sqrt{2\pi}\tilde{\alpha}_{s,k}}\exp\big[-\frac{1}{2}(\frac{\tilde{\beta}_{s,k}}{\tilde{\alpha}_{s,k}})^2\big]
\end{aligned} \tag{59}$$

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

$$<(x)_n\Theta_s>_k = \frac{(C_k\boldsymbol{\alpha}_s)_n}{\sqrt{2\pi}\tilde{\alpha}_{s,k}}\exp\big[-\frac{1}{2}(\frac{\tilde{\beta}_{s,k}}{\tilde{\alpha}_{s,k}})^2\big] + (\boldsymbol{\mu}_k)_n<\Theta_s>_k \tag{60}$$

Where,

$$\tilde{\alpha}_{s,k} = \sqrt{\boldsymbol{\alpha}_s^T C_k\boldsymbol{\alpha}_s} \tag{61}$$

$$\tilde{\beta}_{s,k} = \boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k - \beta_s \tag{62}$$

Next we compute the following required average:

$<\boldsymbol{\Theta_s}>_{\boldsymbol{k}}$

$$\begin{aligned}
&= \frac{1}{(2\pi)^{4/2}\big(det(C_k)\big)^{\frac{1}{2}}}\int_{\mathbb{R}^4}\Theta\big(\boldsymbol{\alpha}_s\cdot\boldsymbol{x} - \beta_s\big)\\
&\quad\exp\left(-\frac{1}{2}\big(\boldsymbol{x}-\boldsymbol{\mu}_k\big)^T C_k^{-1}\big(\boldsymbol{x}-\boldsymbol{\mu}_k\big)\right)d\boldsymbol{x}\\
&= \frac{1}{(2\pi)^{4/2}\big(det(C_k)\big)^{\frac{1}{2}}}\int_{\mathbb{R}^4}\Theta\big(\boldsymbol{\alpha}_s\cdot\boldsymbol{x}^{'} + \boldsymbol{\alpha}_s\cdot\boldsymbol{\mu}_k - \beta_s\big)\\
&\quad\exp\left(-\frac{1}{2}\boldsymbol{x}^{'T}C_k^{-1}\boldsymbol{x}^{'}\right)d\boldsymbol{x}^{'}
\end{aligned}$$

$$= \frac{1}{(2\pi)^2} \int_{\mathbb{R}^4} \Theta\big(\boldsymbol{\alpha}_s C_k^{\frac{1}{2}} \boldsymbol{y} + \boldsymbol{\alpha}_s \cdot \boldsymbol{\mu}_k - \beta_s\big) \exp\left(-\frac{1}{2}\boldsymbol{y}^2 \right) d\boldsymbol{y}$$

$$= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \Theta\left(\|C_k^{\frac{1}{2}} \boldsymbol{\alpha}_s\| \tilde{y} + \boldsymbol{\alpha}_s \cdot \boldsymbol{\mu}_k - \beta_s \right) \exp\big[-\frac{1}{2}\tilde{y}^2 \big] d\tilde{y}$$

$$= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \Theta\big(\tilde{\alpha}_{s,k}\tilde{y} + \tilde{\beta}_{s,k}\big) \exp\big[-\frac{1}{2}\tilde{y}^2 \big] d\tilde{y} \tag{63}$$

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

$$= \frac{1}{\sqrt{2\pi}} \int_{-\frac{\tilde{\beta}_{s,k}}{\tilde{\alpha}_{s,k}}}^{\infty} \exp\big[-\frac{1}{2}\tilde{y}^2 \big] d\tilde{y}$$

$$= 1 - \Phi\big(-\frac{\tilde{\beta}_{s,k}}{\tilde{\alpha}_{s,k}} \big)$$

$$= \Phi\big(\frac{\tilde{\beta}_{s,k}}{\tilde{\alpha}_{s,k}}\big) \tag{64}$$

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

$$< \Theta_s >_k = \Phi\big(\frac{\tilde{\beta}_{s,k}}{\tilde{\alpha}_{s,k}}\big)$$

$$< (\boldsymbol{x})_n \Theta_s >_k = \frac{(C_k \boldsymbol{\alpha}_s)_n}{\sqrt{2\pi}\tilde{\alpha}_{s,k}} \exp\big[-\frac{1}{2}(\frac{\tilde{\beta}_{s,k}}{\tilde{\alpha}_{s,k}})^2 \big] + (\boldsymbol{\mu}_k)_n \Phi\big(\frac{\tilde{\beta}_{s,k}}{\tilde{\alpha}_{s,k}}\big) \tag{65}$$

Where,
$\tilde{\alpha}_{s,k} = \sqrt{\boldsymbol{\alpha}_s^T C_k \boldsymbol{\alpha}_s}$
$\tilde{\beta}_{s,k} = \boldsymbol{\alpha}_s \cdot \boldsymbol{\mu}_k - \beta_s$