ADAPTIVE KERNEL SELF-ORGANIZING MAPS USING INFORMATION THEORETIC LEARNING

By
RAKESH CHALASANI

A THESIS PRESENTED TO THE GRADUATE SCHOOL OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE UNIVERSITY OF FLORIDA 2010
To my brother and sister-in-law, wishing them a happy married life.
ACKNOWLEDGMENTS

I would like to thank Dr. José C. Príncipe for his guidance throughout the course of this research. His constant encouragement and helpful suggestions have shaped this work to its present form. I would also like to thank him for the motivation he gave me and things that I have learned through him will be with me forever.

I would also like to thank Dr. John M. Shea and Dr. Clint K. Slatton for being on the committee and for their valuable suggestions.

I am thankful to my colleagues at the CNEL lab for their suggestions and discussions during the group meetings. I would also like to thank my friends at the UF who made my stay here such a memorable one.

Last but not the least, I would like to thank my parents, brother and sister-in-law for all the support they gave me during some of the tough times I had been through.
# TABLE OF CONTENTS

| ACKNOWLEDGMENTS | 4 |
| LIST OF TABLES | 7 |
| LIST OF FIGURES | 8 |
| ABSTRACT | 9 |

## CHAPTER

1 **INTRODUCTION**
   1.1 **Background**
   1.2 **Motivation and Goals**
   1.3 **Outline**
2 **SELF-ORGANIZING MAPS**
   2.1 **Kohonen’s Algorithm**
   2.2 **Energy Function and Batch Mode**
   2.3 **Other Variants**
3 **INFORMATION THEORETIC LEARNING AND CORRENTROPY**
   3.1 **Information Theoretic Learning**
      3.1.1 **Renyi’s Entropy and Information Potentials**
      3.1.2 **Divergence**
   3.2 **Correntropy and its Properties**
4 **SELF-ORGANIZING MAPS WITH THE CORRENTROPY INDUCED METRIC**
   4.1 **On-line Mode**
   4.2 **Energy Function and Batch Mode**
   4.3 **Results**
      4.3.1 **Magnification Factor of the SOM-CIM**
      4.3.2 **Maximum Entropy Mapping**
5 **ADAPTIVE KERNEL SELF ORGANIZING MAPS**
   5.1 **The Algorithm**
      5.1.1 **Homoscedastic Estimation**
      5.1.2 **Heteroscedastic Estimation**
   5.2 **SOM-CIM with Adaptive Kernels**
6 **APPLICATIONS**
   6.1 **Density Estimation**

---

5
6.2 Principal Surfaces and Clustering ........................................ 47
  6.2.1 Principal Surfaces .................................................. 47
  6.2.2 Avoiding Dead Units ............................................... 49
6.3 Data Visualization ....................................................... 51

7 CONCLUSION ...................................................................... 54

APPENDIX

A MAGNIFICATION FACTOR OF THE SOM WITH MSE .................. 56
B SOFT TOPOGRAPHIC VECTOR QUANTIZATION USING CIM ............ 59
  B.1 SOM and Free Energy .................................................. 59
  B.2 Expectation Maximization ............................................. 59

REFERENCES ........................................................................ 62

BIOGRAPHICAL SKETCH ..................................................... 65
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-1</td>
<td>The information content and the mean square quantization error for various</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>value of $\sigma$ in SOM-CIM and SOM with MSE is shown. $l_{max} = 3.2189$</td>
<td></td>
</tr>
<tr>
<td>5-1</td>
<td>The information content and the mean square quantization error for</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>homoscedastic and heteroscedastic cases in SOM-CIM. $l_{max} = 3.2189$</td>
<td></td>
</tr>
<tr>
<td>6-1</td>
<td>Comparison between various methods as density estimators.</td>
<td>47</td>
</tr>
<tr>
<td>6-2</td>
<td>Number of dead units yielded for different datasets when MSE and CIM are</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>used for mapping.</td>
<td></td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>The SOM grid with the neighborhood range.</td>
<td>14</td>
</tr>
<tr>
<td>3-1</td>
<td>Surface plot CIM(X,0) in 2D sample space. (kernel width is 1)</td>
<td>24</td>
</tr>
<tr>
<td>4-1</td>
<td>The unfolding of mapping when CIM is used.</td>
<td>28</td>
</tr>
<tr>
<td>4-2</td>
<td>The scatter plot of the weights at convergence when the neighborhood is rapidly decreased.</td>
<td>29</td>
</tr>
<tr>
<td>4-3</td>
<td>The figure shows the plot between $\ln(r)$ and $\ln(w)$ for different values of $\sigma$.</td>
<td>31</td>
</tr>
<tr>
<td>4-4</td>
<td>The figures shows the input scatter plot and the mapping of the weights at the final convergence.</td>
<td>33</td>
</tr>
<tr>
<td>4-5</td>
<td>Negative log likelihood of the input versus the kernel bandwidth.</td>
<td>34</td>
</tr>
<tr>
<td>5-1</td>
<td>Kernel adaptation in homoscedastic case.</td>
<td>38</td>
</tr>
<tr>
<td>5-2</td>
<td>Kernel adaptation in heteroscedastic case.</td>
<td>39</td>
</tr>
<tr>
<td>5-3</td>
<td>Magnification of the map in homoscedastic and heteroscedastic cases.</td>
<td>42</td>
</tr>
<tr>
<td>5-4</td>
<td>Mapping of the SOM-CIM with heteroscedastic kernels and $\rho = 1$.</td>
<td>43</td>
</tr>
<tr>
<td>5-5</td>
<td>The scatter plot of the weights at convergence in case of homoscedastic and heteroscedastic kernels.</td>
<td>44</td>
</tr>
<tr>
<td>6-1</td>
<td>Results of the density estimation using SOM.</td>
<td>46</td>
</tr>
<tr>
<td>6-2</td>
<td>Clustering of the two-crescent dataset in the presence of outlier noise.</td>
<td>48</td>
</tr>
<tr>
<td>6-3</td>
<td>The scatter plot of the weights showing the dead units when CIM and MSE are used to map the SOM.</td>
<td>49</td>
</tr>
<tr>
<td>6-4</td>
<td>The U-matrix obtained for the artificial dataset that contains three clusters.</td>
<td>51</td>
</tr>
<tr>
<td>6-5</td>
<td>The U-matrix obtained for the Iris and blood transfusion datasets.</td>
<td>52</td>
</tr>
<tr>
<td>B-1</td>
<td>The unfolding of the SSOM mapping when the CIM is used as the similarity measure.</td>
<td>60</td>
</tr>
</tbody>
</table>
Abstract of Thesis Presented to the Graduate School of the University of Florida in Partial Fulfillment of the Requirements for the Degree of Master of Science

ADAPTIVE KERNEL SELF-ORGANIZING MAPS USING INFORMATION THEORETIC LEARNING

By

Rakesh Chalasani

May 2010

Chair: José Carlos Príncipe
Major: Electrical and Computer Engineering

Self organizing map (SOM) is one of the popular clustering and data visualization algorithm and evolved as a useful tool in pattern recognition, data mining, etc., since it was first introduced by Kohonen. However, it is observed that the magnification factor for such mappings deviates from the information theoretically optimal value of 1 (for the SOM it is 2/3). This can be attributed to the use of the mean square error to adapt the system, which distorts the mapping by over sampling the low probability regions.

In this work, the use of a kernel based similarity measure called correntropy induced metric (CIM) for learning the SOM is proposed and it is shown that this can enhance the magnification of the mapping without much increase in the computational complexity of the algorithm. It is also shown that adapting the SOM in the CIM sense is equivalent to reducing the localized cross information potential and hence, the adaptation process as a density estimation procedure. A kernel bandwidth adaptation algorithm in case of the Gaussian kernels, for both homoscedastic and heteroscedastic components, is also proposed using this property.
CHAPTER 1
INTRODUCTION

1.1 Background

The topographic organization of the sensory cortex is one of the complex phenomenons in the brain and is widely studied. It is observed that the neighboring neurons in the cortex are stimulated from the inputs that are in the neighborhood in the input space and hence, called as neighborhood preservation or topology preservation. The neural layer acts as a topographic feature map, if the locations of the most excited neurons are correlated in a regular and continuous fashion with a restricted number of signal features of interest. In such a case, the neighboring excited locations in the layer corresponds to stimuli with similar features (Ritter et al., 1992).

Inspired from such a biological plausibility, but not intended to explain it, Kohonen (1982) has proposed the Self Organizing Maps (SOM) where the continuous inputs are mapped into discrete vectors in the output space while maintaining the neighborhood of the vector nodes in a regular lattice. Mathematically, the Kohonen’s algorithm (Kohonen, 1997) is a neighborhood preserving vector quantization tool working on the principle of winner-take-all, where the winner is determined as the most similar node to the input at that instant of time, also called as the best matching unit (BMU). The center piece of the algorithm is to update the BMU and its neighborhood nodes concurrently. By performing such a mapping the input topology is preserved on to the grid of nodes in the output. The details of the algorithm are explained in Chapter 2.

The neural mapping in the brain also performs selective magnification of the regions of interest. Usually these regions of interest are the ones that are often excited. Similar to the neural mapping in the brain, the SOM also magnifies the regions that are often excited (Ritter et al., 1992). This magnification can be explicitly expressed as a power law between the input data density \( P(v) \) and the weight vector density \( P(w) \) at the time of convergence. The exponent is called as the magnification factor or magnification.
exponent (Villmann and Claussen, 2006). A faithful representation of the data by the weights can happen only when this magnification factor is 1.

It is shown that if the mean square error (MSE) is used as the similarity measure to find the BMU and also to adapt a 1D-1D SOM or in case of separable input, then the magnification factor is $2/3$ (Ritter et al., 1992). So, such a mapping is not able to transfer optimal information from the input data to the weight vectors.

Several variants of the SOM are proposed that can make the magnification factor equal to 1. They are discussed in the following chapters.

1.2 Motivation and Goals

The reason for the sub-optimal mapping of the traditional SOM algorithm can be attributed to the use of euclidian distance as the similarity measure. Because of the global nature of the MSE cost function, the updating of the weight vectors is greatly influenced by the outliers or the data in the low probability regions. This leads to over sampling of the low probability regions and under sampling the high probability regions by the weight vectors. Here, to overcome this, a localized similarity measure called correntropy induced metric is used to train the SOM, which can enhance the magnification of the SOM.

Correntropy is defined as a generalized correlation function between two random variables in a high dimensional feature space (Santamaria et al., 2006). This definition induces a reproducing kernel Hilbert space (RKHS) and the euclidian distance between two random variables in this RKHS is called as Correntropy Induced Metric (CIM) (Liu et al., 2007). When Gaussian kernel induces the RKHS it is observed that the CIM has a strong outlier’s rejection capability in the input space. It also includes all the even higher order moments of the difference between the two random variables in the input space.

The goal of this thesis is to show that these properties of the CIM can help to enhance the magnification of the SOM. Also finding the relation between this and the other information theoretic learning (ITL) (Principe, 2010) based measures can help
to understand the use of the CIM in this context. It can also help to adapt the free parameter in the correntropy, the kernel bandwidth, such that it can give an optimal solution.

1.3 Outline

This thesis comprises of 7 chapters. In Chapter 2 the traditional self organizing maps algorithm, the energy function associated with the SOM and its batch mode variant are discussed. Some other variants of the SOM are also briefly discussed. Chapter 3 deals with the information theoretic learning followed by the correntropy and its properties. In Chapter 4 the use of the CIM in the SOM is introduced and also the advantage of using the CIM is shown through some examples. In Chapter 5 a kernel bandwidth adaptation algorithm is proposed and is extended to the SOM with the CIM (SOM-CIM). The advantages of using the SOM-CIM are shown in Chapter 6; where it is applied to some of the important application of the SOM like density estimation, clustering and data visualization. The thesis is concluded in Chapter 7.
CHAPTER 2
SELF-ORGANIZING MAPS

Self organizing map is a biologically inspired tool with primary emphasis on the topological order. It is inspired from the modeling of the brain in which the information propagates through the neural layers while preserving the topology of one layer in the other. Since they were first introduced by Kohonen (1997), they have evolved as useful tools in many applications such as visualization of high dimensional data in low dimensional grid, data mining, clustering, compression, etc. In the section 2.1 the Kohonen’s algorithm is briefly discussed.

2.1 Kohonen’s Algorithm

Kohonen’s algorithm is inspired from vector quantization, in which a group of inputs are quantized by a few weight vectors called as nodes. However, in addition to the quantization of the inputs, here the nodes are arranged in regular low dimensional grid and the order of the grid is maintained throughout the learning. Hence, the distribution of the input data in the high dimensional spaces can be preserved on the low dimensional grid (Kohonen, 1997).

Before going further into the details of the algorithm, please note that the following notation is used throughout this thesis: The input distribution \( V \in \mathbb{R}^d \) is mapped as function \( \Phi: V \rightarrow A \), where \( A \) in a lattice of \( M \) neurons with each neuron having a weight vector \( w_i \in \mathbb{R}^d \) where \( i \) are lattice indices.

The Kohonen’s algorithm is described in the following points:

- At each instant \( t \), a random sample, \( v \), from the input distribution \( V \) is selected and the best matching unit (BMU) corresponding to it is obtained using

  \[
  r = \arg \min_s D(w_s, v) \tag{2–1}
  \]

  where node \( r \) is considered as the index of winning node. Here ‘\( D \)’ is any similarity measure that is used to compare the closeness between the two vectors.
Once the winner is obtained the weights of all the nodes should be updated in such a way that the local error given by (2–2) should be minimized.

\[ e(v, w_i) = \sum_s h_{is} D(v - w_s) \]  \hspace{1cm} (2–2)

Here \( h_{is} \) is the neighborhood function; a non-increasing function of the distance between the winner and any other nodes in the lattice. To avoid local minima, as shown by Ritter et al. (1992), it is considered as a convex function, like the middle region of Gaussian function, with a large range at the start and is gradually reduced to \( \delta \).

If the similarity measure considered is the euclidian distance, \( D(v - w_s) = \|v - w_s\|^2 \), then the on-line updating rule for the weights is obtained by taking the derivative and minimizing (2–2). It comes out to be

\[ w_s(n + 1) = w_s(n) + \epsilon h_{is}(v(n) - w_s(n)) \]  \hspace{1cm} (2–3)

Figure 2-1. The SOM grid with the neighborhood range. The circles indicate the number of nodes that correlate in the same direction as the winner; here it is node (4,4).
As discussed earlier, one of the important properties of the SOM is topological preservation and it is the neighborhood function that is responsible for this. The role played by neighborhood function can be better summarized as described in (Haykin, 1998). The reason for large neighborhood is to correlate the direction of weight updates of large number of weights around the BMU, \( r \). As the range decreases, so does the number of neurons correlated in the same direction (Figure 2-1). This correlation ensures that the similar inputs are mapped together and hence, the topology is preserved.

### 2.2 Energy Function and Batch Mode

Erwin et al. (1992) have shown that in case of finite set of training pattern the energy function of the SOM is highly discontinues and in case of continuous inputs the energy function does not exist. It is clear that things go wrong at the edge of the Voronoi regions where the input is equally closer to two nodes when the best matching unit (BMU) is selected using \( 2–1 \). To overcome this Heskes (1999) has proposed that by slightly changing the selection of the BMU there can be a well defined energy function for the SOM.

If the BMU is selected using \( 2–4 \) instead of \( 2–1 \) then the function over which the SOM is minimized is given by \( 2–5 \) (in discrete case). This is also called as the *local error*.

\[
\begin{align*}
\mathbf{r} & = \arg \min_s \sum_t h_{ts} D(\mathbf{v}(n) - \mathbf{w}_t) \quad (2–4) \\
E(W) & = \sum_s h_{ts} D(\mathbf{v}(n) - \mathbf{w}_s) \quad (2–5)
\end{align*}
\]

The overall cost function when a finite number of input samples, say \( N \), are consider is given by

\[
E(W) = \sum_n \sum_s h_{ts} D(\mathbf{v}(n) - \mathbf{w}_s) \quad (2–6)
\]
To find the batch mode update rule, we can take the derivative of $E(W)$ w.r.t $w_s$ and find the value of the weight vectors at the stationary point of the gradient. If the euclidean distance is used then the batch mode update rule is

$$w_s(n+1) = \frac{\sum_n h_{is}x_n}{\sum_n h_{is}}$$

(2-7)

2.3 Other Variants

Contrary to what is assumed in case of the SOM-MSE, the weight density at convergence, also defined as the inverse of the magnification factor, is not proportional to the input density. It is shown by Ritter et al. (1992) that in a continuum mapping, i.e., having infinite neighborhood node density, in a 1D map (i.e, a chain) developed in a one dimensional input space or multidimensional space which are separable, the weight density $P(w) \propto P(v)^{2/3}$ with a non-vanishing neighborhood function. (For proof please refer Appendix A.) When a discrete lattice is used there is a correction in the relation and is given by

$$p(w) = p(v)^{\alpha}$$

(2-8)

with $\alpha = \frac{2}{3} - \frac{1}{3\sigma_h^2 + 3(\sigma_h + 1)^2}$

where $\sigma_h$ is the neighborhood range in case of rectangular function. There are several other methods (Van Hulle, 2000) proposed with the change in the definition of neighborhood function but the mapping is unable to produce an optimal mapping, i.e, magnification of 1. It is observed that the weights are always over sampling the low probability regions and under sampling the high probability regions.

The reason for such a mapping can be attributed to the global nature of the MSE cost function. When the euclidean distance is used, the points at the tail end of the input distribution have a greater influence on overall distortion. This is the reason why the use of the MSE as a cost function is suitable only for thin tail distributions like the Gaussian.
distribution. This property of the euclidean distance pushes the weights into regions of low probability and hence, over sampling that region.

By slightly modifying the updating rule, Bauer and Der (1996) and Claussen (2005) have proposed different methods to obtain a mapping with magnification factor of 1. Bauer and Der (1996) have used the local input density at the weights to adaptively control the step size of the weight update. Such mapping is able to produce the optimal mapping in the same continuum conditions as mentioned earlier but needs to estimate the unknown weight density at a particular point making it unstable in higher dimensions (Van Hulle, 2000). While, the method proposed by Claussen (2005) is not able to produce a stable result in case of high dimensional data.

A completely different approach is taken by Linsker (1989) where mutual information between the input density and the weight density is maximized. It is shown that such a learning based on the information theoretic cost function leads to an optimal solution. But the complexity of the algorithm makes it impractical and, strictly speaking, it is applicable to only for a Gaussian distribution. Furthermore, Van Hulle (2000) has proposed an another information theory based algorithm based on the Bell and Sejnowski (1995)’s Infomax principle where the differential entropy of the output nodes is maximized.

In the recent past, inspired from the use of kernel Hilbert spaces by Vapnik (1995), several kernel based topographic mapping algorithms are proposed (Andras, 2002; Graepel, 1998; MacDonald, 2000). Graepel (1998) has used the theory of deterministic annealing (Rose, 1998) to develop a new self organizing network called the soft topographic vector quantization (STVQ). A kernel based STVQ (STMK) is also proposed where the weights are considered in the feature space rather than the input space. To overcome this, Andras (2002) proposed a kernel-Kohonen network in which the the input space is transformed, both the inputs and the weights, into a high dimensional reproducing kernel Hilbert space (RKHS) and used the euclidian distance in the high

17
dimensional space as the cost function to update the weights. It is analyzed in the context of classification and the idea comes from the theory of non-linear support vector machines (Haykin, 1998) which states -

If the boundary separating the two classes is not linear, then there exist a transformation of the data in another space in which the two classes are linearly separable.

But Lau et al. (2006) showed that the type of the kernel strongly influences the classification accuracy and it is also shown that kernel-Kohonen network does not always outperform the basic kohonen network. Moreover, Yin (2006) has compared KSOM with the self organizing mixture networks (SOMN) (Yin and Allinson, 2001) and showed that KSOM is equivalent to SOMN, and in turn the SOM itself, but failed to establish why KSOM gives a better classification than the SOM.

Through this thesis, we try to use the idea of Correntropy Induced Metric (CIM) and show how the type of the kernel function influences the mapping. We also show that the kernel bandwidth plays an important role in the map formation. In the following chapter we briefly discuss the theory of information theoretic learning (ITL) and Correntropy before going into the details of the proposed algorithm.
CHAPTER 3
INFORMATION THEORETIC LEARNING AND CORRENTROPY

3.1 Information Theoretic Learning

The central idea of the information theoretic learning proposed by Principe et al. (2000) is non-parametrically estimating the measures like entropy, divergence, etc.

3.1.1 Renyi’s Entropy and Information Potentials

Renyi’s $\alpha$-order entropy is defined as

$$H_\alpha(X) = \frac{1}{1 - \alpha} \log \int f_X^{\alpha}(x)$$  \hspace{1cm} (3–1)

A special case with $\alpha = 2$ is considered and is called as the Renyi’s quadratic entropy. It is defined over a random variable $X$ as

$$H_2(X) = -\log \int f_X^2(x)dx$$  \hspace{1cm} (3–2)

The pdf of $X$ is estimated using the famous Parzen density estimation (Parzen, 1962) and is given by

$$\hat{f}_x(x) = \frac{1}{N} \sum_n \kappa_\sigma(x - x_n)$$  \hspace{1cm} (3–3)

where $x_n; n = 1, 2, ..., N$ are $N$ samples obtained from the pdf of $X$. Popularly, a Gaussian kernel defined as

$$G_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{(x)^2}{2\sigma^2} \right)$$  \hspace{1cm} (3–4)

is used for the Parzen estimation. In case of multidimensional density estimation, the use of joint kernels of the type

$$G_{\sigma}(x) = \prod_d G_{\sigma_d}(x(d))$$  \hspace{1cm} (3–5)

where $D$ is the number of dimensions of the input vector $x$, is suggested.

Substituting 3–3 in place of the density functions in 3–2, we get

$$H_2(X) = -\log \int \left( \frac{1}{N} \sum_n G_\sigma(x - x_n) \right)^2 dx$$
\[
\begin{align*}
&= -\log \int \frac{1}{N^2} \sum_{n}^{N} \sum_{m}^{N} G_{\sigma}(x - x_n) G_{\sigma}(x - x_m) dx \\
&= -\log \frac{1}{N^2} \sum_{n}^{N} \sum_{m}^{N} \int G_{\sigma}(x - x_n) G_{\sigma}(x - x_m) dx \\
&= -\log \frac{1}{N^2} \sum_{n}^{N} \sum_{m}^{N} G_{\sigma \sqrt{2}}(x_n - x_m) \\
&= -\log IP(x)
\end{align*}
\]

where

\[
IP(x) = \frac{1}{N^2} \sum_{n}^{N} \sum_{m}^{N} G_{\sigma}(x_n - x_m)
\]

(3–7)

Here 3–7 is called as the information potential of X. It defines the state of the particles in the input space and interactions between the information particles. It is also shown that the Parzen pdf estimation of X is equivalent to the information potential field (Deniz Erdogmus, 2002).

Another measure which can estimate the interaction between the points of two random variables in the input space is the cross-information potential (CIP) (Principe et al., 2000). It is defined as an integral of the product of two probability density functions which characterizes similarity between two random variables. It determines the effect of the potential created by \( g_y \) at a particular location in the space defined by the \( f_x \) (or vice versa); where \( f \) and \( g \) are the probability density function.

\[
CIP(X, Y) = \int f(x) g(x) dx
\]

(3–8)

Using the Parzen density estimation again with the Gaussian kernel, we get

\[
CIP(X, Y) = \int \frac{1}{MN} \sum_{n}^{N} \sum_{m}^{M} G_{\sigma_x}(u - x_n) G_{\sigma_y}(u - y_m) du
\]

(3–9)

Similar to the information potential, we get

\[
CIP(X, Y) = \frac{1}{NM} \sum_{n}^{N} \sum_{m}^{M} G_{\sigma_x}(x_n - y_m)
\]

(3–10)
with $\sigma^2 = \sigma^2_f + \sigma^2_g$.

3.1.2 Divergence

The dissimilarity between two density functions can be quantified using the divergence measure. Rényi (1961) has proposed an $\alpha$-order divergence measure for which the popular KL-divergence is a special case. The Rényi’s divergence between the two density function $f$ and $g$ is given by

$$D_\alpha(f, g) = \frac{1}{1-\alpha} \log \int f(x) \left( \frac{f(x)}{g(x)} \right)^{\alpha-1} dx$$

(3–11)

Again, by varying the value of $\alpha$ the user can set the divergence varying different order. As a limiting case when $\alpha \to 1$, $D_\alpha \to D_{KL}$; where $D_{KL}$ refers to KL-Divergence and is given by

$$D_{KL}(f, g) = \int f(x) \log \left( \frac{f(x)}{g(x)} \right) dx$$

(3–12)

Using the similar approach used to calculate the Rényi’s entropy, to estimate the $D_{KL}$ from the input samples the density functions can be substituted using the Parzen estimation 3–3. If $f$ and $g$ are probability density functions (pdf) of the two random variables $X$ and $Y$, then

$$D_{KL}(f, g) = \int f_x(u) \log(f_x(u)) du - \int f_x(u) \log(g_y(u)) du$$

$$= E_x[\log(f_x)] - E_x[\log(g_y)]$$

$$= E_x \left[ \log \left( \sum_i G_x(x - x(i)) \right) \right] - E_x \left[ \log \left( \sum_j G_y(x - y(j)) \right) \right]$$

(3–13)

The first term in 3–13 is equivalent to the Shannon’s entropy of $X$ and the second term is equivalent to the cross-entropy between the $X$ and $Y$.

Several other divergence measures are proposed using the information potential(IP) like the Cauchy-Schwarz divergence and quadratic divergence but are not discussed here. Interested readers can refer to (Principe, 2010).
3.2 Correntropy and its Properties

Correntropy is a generalized similarity measure between two arbitrary random variables $X$ and $Y$ defined by (Liu et al., 2007).

$$V_\sigma(X, Y) = E(\kappa_\sigma(X - Y)) \quad (3-14)$$

If $\kappa$ is a kernel that satisfies the Mercer’s Theorem (Vapnik, 1995), then it induces a reproducing kernel Hilbert space (RKHS) called as $\mathcal{H}_\kappa$. Hence, it can also be defined as the dot product of the two random variables in the feature space, $\mathcal{H}_\kappa$.

$$V(X, Y) = E[<\phi(x)>, <\phi(y)>] \quad (3-15)$$

where $\phi$ is a non-linear mapping from the input space to the feature space such that $\kappa(x, y) = [<\phi(x)>, <\phi(y)>]$; where $<.,.>$ defines inner product.

Here we use the Gaussian kernel

$$G_\sigma(x, y) = \frac{1}{(\sqrt{2\pi}\sigma)^d}e^{\exp(-\frac{||x - y||^2}{2\sigma^2})} \quad (3-16)$$

where $d$ is the input dimensions; since it is most popularly used in the information theoretic learning (ITL) literature. Another popular kernel used is the Cauchy kernel which is given by

$$C_\sigma(x, y) = \frac{\sigma}{\sigma^2 + ||x - y||^2} \quad (3-17)$$

In practice, only a finite number of samples of the data are available and hence, correntropy can be estimated as

$$\hat{V}_{N,\sigma} = \frac{1}{N} \sum_{i=1}^{N} \kappa_\sigma(x_i - y_i) \quad (3-18)$$

Some of the properties of correntropy that are important in this thesis are discussed here and detailed analysis can be obtained at (Liu et al., 2007; Santamaria et al., 2006).

Property 1: Correntropy involves all the even order moments of the random variable $E = X - Y$. 
Expanding the Gaussian kernel in 3–14 using the Taylor’s series, we get

\[
V(X, Y) = \frac{1}{\sqrt{2\pi}\sigma} \sum_{n=0}^{\infty} \frac{(-1)^n}{2^n n!} E \left[ \frac{(X - Y)^{2n}}{\sigma^{2n}} \right]
\]  

(3–19)

As \( \sigma \) increases the higher order moments decay and second order moments dominate. For a large \( \sigma \) correntropy approaches correlation.

**Property 2:** Correntropy, as a sample estimator, induces a metric called Correntropy induced metric (CIM) in the sample space. Given two sample vectors \( X(x_1, x_2, \ldots, x_N) \) and \( Y(y_1, y_2, \ldots, y_N) \), CIM is defined as

\[
CIM(X, Y) = (\kappa_{\sigma}(0) - \hat{V}_{\sigma}(X, Y))^{1/2} = \left( \frac{1}{N} \sum_{n} \kappa_{\sigma}(0) - \kappa_{\sigma}(x_n - y_n) \right)^{1/2}
\]  

(3–20)

(3–21)

The CIM can also be viewed as the euclidian distance in the feature space if the kernel is of the form \( \|x - y\| \). Consider two random variables \( X \) and \( Y \) transformed into the feature space using a non-linear mapping \( \Phi \), then

\[
E[\|\phi(x) - \phi(y)\|^2] = E[\kappa_{\sigma}(x, x) + \kappa_{\sigma}(y, y) - 2\kappa_{\sigma}(x, y)]
\]

\[
= E[2\kappa_{\sigma}(0) - 2\kappa_{\sigma}(x, y)]
\]

\[
= \frac{2}{N} \sum_{n} \left( \kappa_{\sigma}(0) - \kappa_{\sigma}(x_n - y_n) \right)
\]  

(3–22)

Except for the additional multiplying factor of 2, 3–22 is similar to 3–21.

Since \( \phi \) is a non-linear mapping from the input to the feature space, the CIM induces a non-linear similarity measure in the input space. It has been observed that the CIM behaves like an L2 norm when the two vectors are close, as L1 norm outside the L2 norm and as they go farther apart it becomes insensitive to distance between the two vectors (L0 norm). And the extent of the space over which the CIM acts as L2 or L0 norm is directly related to the kernel bandwidth, \( \sigma \). This unique property of CIM localizes the similarity measure and is very helpful in rejecting the outliers. In this regard it, is very different from the MSE which provides a global measure.
Figure 3-1. Surface plot CIM(X,0) in 2D sample space. (kernel width is 1)

Figure 3-1 demonstrates the non-linear nature of the surface of the CIM, with N=2. Clearly the shape of L2 norm depends on the kernel function. Also the kernel bandwidth determines the extent L2 regions. As we will see later, these parameters play an important role in determining the quality of the final output.
CHAPTER 4
SELF-ORGANIZING MAPS WITH THE CORRENTROPY INDUCED METRIC

As discussed in the previous chapter correntropy induces a non-linear metric in the input space called as Correntropy Induced Metric (CIM). Here we show that the CIM can be used as a similarity measure in the SOM to determine the winner and also to update the weight vectors.

4.1 On-line Mode

If \( \mathbf{v}(n) \) is considered as the input vector of the random variable \( V \) at an instant \( n \), then the best matching unit (BMU) can be obtained using the CIM as

\[
\mathbf{r} = \arg\min_{s} CIM(\mathbf{v}(n), \mathbf{w}_s)_{N=1} = \arg\min_{s}(\kappa_\sigma(0) - \kappa_\sigma(\|\mathbf{v}(n) - \mathbf{w}_s\|))
\]  

(4–1)

(4–2)

where \( \mathbf{r} \) is consider as the BMU at instant \( n \) and \( \sigma \) is predetermined.

According to Kohonen’s algorithm, if the CIM is used as the similarity measure then the local error that needs to be minimized to obtain an ordered mapping is given by

\[
e(\mathbf{w}_r, \mathbf{v}) = \sum_{s=1}^{M} h_{rs}(\kappa_\sigma(0) - \kappa_\sigma(\|\mathbf{v}(n) - \mathbf{w}_s\|))
\]  

(4–3)

For performing gradient descent over the local error for updating the weights, take the derivative of 4–3 with respect to \( \mathbf{w}_s \) and update the weights using the gradient. The update rule comes out to be

\[
\mathbf{w}_s(n+1) = \mathbf{w}_s(n) - \eta \Delta \mathbf{w}_s
\]  

(4–4)

If the Gaussian function is used then the gradient is

\[
\Delta \mathbf{w}_s = -h_{rs}G_\sigma(\|\mathbf{v}(n) - \mathbf{w}_s\|)(\mathbf{v}(n) - \mathbf{w}_s)
\]  

(4–5)

In case of Cauchy kernel, it is

\[
\Delta \mathbf{w}_s = -h_{rs}\frac{1}{(\sigma^2 + \|\mathbf{v}(n) - \mathbf{w}_s\|^2)}(\mathbf{v}(n) - \mathbf{w}_s)
\]  

(4–6)
The implicit $\sigma$ term in 4–5 and 4–6 is combined with the learning rate $\eta$, a free parameter.

When compared to the update rule obtained using the MSE (2–3), the gradient in both the cases have an additional scaling factor. This additional scaling factor, whose value is small when $\|v(n) - w_s\|$ is large, in the updating rule points out the strong outliers rejection capability of the CIM. This property of the CIM is able to resolve one of the key problems associated with using MSE, i.e., over sampling the low probability regions of the input distribution. Since the weights are less influenced by the inputs in the low probability regions, the SOM with the CIM (SOM-CIM) can give a better magnification. It should be noted that since the influence of the outliers depends on the shape and extent of the L2-norm and L0-norm regions of the CIM, the magnification factor in turn is dependent on the type of the kernel and its bandwidth, $\sigma$.

Another property of the CIM (with a Gaussian kernel), that influences the magnification factor, is the presence of higher order moments. According to Zador (1982) the order of the error directly influences the magnification factor of the mapping. Again, since the kernel bandwidth, $\sigma$, determines the influence of the higher order moments (refer to section 3.2) the choice of $\sigma$ plays an important role in the formation of the final mapping.

### 4.2 Energy Function and Batch Mode

If the CIM is used for determining the winner and to update the weights, then according to Heskes (1999) an energy function over which the SOM can be adapted exist only when the winner is determined using the local error rather than just a metric. Hence, if the BMU is obtained by using

$$ r = \arg\min_s \sum_t h_{ts}\text{CIM}(v(n) - w_t) $$

(4–7)

then the SOM-CIM is adapted by minimizing the following energy function.

$$ e(W) = \sum_{s}^{M} h_{ts}\text{CIM}(v(n) - w_s) $$

(4–8)
In the batch mode, when a finite number of input samples are present the overall cost function becomes

\[ E_{CIM}(W) = \sum_{n} \sum_{s} h_{is} CIM(v(n) - w_s) \]  \hspace{1cm} (4–9)

Similar to the batch mode update rule obtained while using the MSE, we find the weight vectors at stationary point of the gradient of the above energy function, \( E_{CIM}(W) \). In case of the Gaussian kernel

\[ \frac{\partial E_{CIM}(W)}{\partial w_s} = - \sum_{n} h_{is} G(\sigma)(v(n) - w_s)(v(n) - w_s) = 0 \]  \hspace{1cm} (4–10)

\[ \Rightarrow w_s = \frac{\sum_{n} h_{is} G(\sigma)(v(n) - w_s)v(n)}{\sum_{n} h_{is} G(\sigma)(v(n) - w_s)} \]  \hspace{1cm} (4–11)

The update here comes out to be the so-called iterative fixed point update rule, indicating that the weights are updated iteratively and can not reach minima in one iteration, like it does with the MSE.

And in case of the Cauchy kernel, the update rule is

\[ w_s = \frac{\sum_{n} h_{is} C(\sigma)(v(n) - w_s)v(n)}{\sum_{n} h_{is} C(\sigma)(v(n) - w_s)} \]  \hspace{1cm} (4–12)

The scaling factor in the denominator and the numerator plays the similar role as described in the previous section 4.1.

Figure 4–1 shows how the map evolves when the CIM is used to map 4000 samples of a uniform distribution between (0,1] on a 7x7 rectangular grid. It shows the scatter plot of the weights at different epochs and the neighborhood relation between the nodes in the lattice are indicated by the lines. Initially, the weights are randomly chosen between (0,1] and the kernel bandwidth is kept constant at 0.1. A Gaussian neighborhood function, given by 4–13, is considered and its range is decreased with the number of epochs using 4–14.

\[ h_{is}(n) = \exp \left( - \frac{(r - s)^2}{2\sigma_s(n)} \right) \]  \hspace{1cm} (4–13)
A Random initial condition

B Epoch = 10

C Epoch = 30

D Epoch = 50

Figure 4-1. The figure shows the scatter plot of the weights along with the neighborhood nodes connected with the lines at various stages of the learning.

\[
\sigma_h(n) = \sigma_h^i \exp \left( -\theta \ast \frac{\sigma_h^i n}{N} \right)
\]

(4–14)

where \( r \) and \( s \) are node indexes, \( \sigma_h \) is the range of the neighborhood function, \( \sigma_h^i \) = initial value of \( \sigma_h \), \( \theta = 0.3 \) and \( N = \) number of iterations/epochs.

In the first few epochs, when the neighborhood is large, all the nodes converge around the mean of the inputs and as the neighborhood decreases with the number of epochs, the nodes start to represent the structure of the data and minimizes the error.

An another important point regrading the kernel bandwidth should be considered here. If the value of \( \sigma \) is small, then during the 'ordering phase' (Haykin, 1998) of the
map the moment of the nodes is restricted because of the small L2 region. So, a slow annealing of the neighborhood function is required. To overcome this, a large $\sigma$ is considered initially, which ensures that the moment of the nodes is not restricted. It is gradually annealed during the ordering phase and kept constant at the desired value during the convergence phase. Figure 4-2 demonstrates this. When the kernel size is kept constant and the neighborhood function is decreased rapidly with $\theta = 1$ in 4–14, a requirement for faster convergence, the final mapping is distorted. On the other hand, when the $\sigma$ is annealed gradually from a large value to a small value, in this case 5 to 0.1, during the ordering phase and kept constant after that, the final mapping is much less distorted. Henceforth, this is the technique used to obtain all the results.

In addition to this, it is interesting if we look back at the cost function (or local error) when the Gaussian kernel is used. It is reproduced here and is given by

$$e_{CIM}(\mathcal{W}) = \sum_{s}^{M} h_{s}(1 - G_{s}(\mathbf{w}_{s}, \mathbf{v}(n)))$$  \hspace{1cm} (4–15)  

$$= \sum_{s}^{M} h_{s} - \sum_{s}^{M} h_{s}G_{s}(\mathbf{w}_{s}, \mathbf{v}(n))$$  \hspace{1cm} (4–16)
The second term in the above equation can be considered as the estimate of the probability of the input using the sum of the Gaussian mixtures centered at the weights; with the neighborhood function considered equivalent to the posterior probability.

\[ p(v(n)/W) \approx \sum_i p(v(n)/w_i) P(w_i) \]  \hspace{1cm} (4–17)

Hence, it can be said that this way of kerneling of the SOM is equivalent to estimating the input probability distribution (Yin, 2006). It can also be considered as the weighted Parzen density estimation (Babich and Camps, 1996) technique; with the neighborhood function acting as the strength associated with each weight vector. This reasserts the argument that the SOM-CIM can increase the magnification factor of the mapping to 1.

4.3 Results

4.3.1 Magnification Factor of the SOM-CIM

As pointed out earlier, the use of the CIM does not allow the nodes to over sample the low probability regions of the input distribution as they do with the MSE. Also the presence of the higher order moments effect the final mapping but it is difficult to determine the exact effect the higher order moments have on the mapping here. Still, it can be expected that the magnification of the SOM can be improved using CIM. To verify this experimentally, we use a setup similar to that used by Ritter et al. (1992) to demonstrate the magnification of the SOM using the MSE.

Here, a one dimensional input space is mapped onto a one dimensional map, i.e., a chain, with 50 nodes where the input is 100,000 instances drawn from the distribution \( f(x) = 2x \). Then the relation between the weights and the node indexes are compared. A Gaussian neighborhood function, given by 4–13, is considered and its range is decreased with the number of iterations using 4–14 with \( \theta = 0.3 \).

If the magnification factor is ‘1’, as in the case of optimal mapping, the relation between the weights and the nodes is \( \ln(w) = \frac{1}{2} \ln(r) \). In case of the SOM with the MSE (SOM-MSE) as the cost function, the magnification is proved to be 2/3 (Appendix.
Figure 4-3. The figure shows the plot between $\ln(r)$ and $\ln(w)$ for different values of $\sigma$. ‘-’ indicates the ideal plot representing the relation between w and r when the magnification is 1 and ‘...’ indicates the ideal plot representing the magnification of 2/3. ‘-+-’ indicates relation obtained with SOM-CIM when $\sigma = 0.03$ and ‘-x-’ indicates when $\sigma = 0.5$.

A) and the relation comes out to be $\ln(w) = \frac{3}{5}\ln(r)$. Figure 4-3 shows that when smaller $\sigma$ is considered for the SOM-CIM then a better magnification can be achieved. As the value of the $\sigma$ increases the mapping resembles the one with the MSE as the cost function. This establishes the nature of the surface of CIM, as discussed in Section 3.2, that with the increase in the value of $\sigma$ the surface tends to behave more like MSE. Actually, it can be said that with the change in the value of the $\sigma$ the magnification factor of the SOM can vary between 2/3 to 1! But an analytical solution that can give a relation between the two is still elusive.
Table 4-1. The information content and the mean square quantization error for various value of $\sigma$ in SOM-CIM and SOM with MSE is shown. $l_{max} = 3.2189$

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean Square Quantization Error</th>
<th>Entropy or Info Cont., $l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSOM kernel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bandwidth, $\sigma$</td>
<td>($\times 10^{-3}$)</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>15.361</td>
<td>3.1301</td>
</tr>
<tr>
<td>Gaussian</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>5.724</td>
<td>3.2101</td>
</tr>
<tr>
<td>0.2</td>
<td>5.082</td>
<td>3.1929</td>
</tr>
<tr>
<td>0.5</td>
<td>5.061</td>
<td>3.1833</td>
</tr>
<tr>
<td>0.8</td>
<td>5.046</td>
<td>3.1794</td>
</tr>
<tr>
<td>1.0</td>
<td>5.037</td>
<td>3.1750</td>
</tr>
<tr>
<td>1.5</td>
<td>5.045</td>
<td>3.1768</td>
</tr>
<tr>
<td>Cauchy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kernel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>5.3305</td>
<td>3.2065</td>
</tr>
<tr>
<td>0.2</td>
<td>5.1399</td>
<td>3.1923</td>
</tr>
<tr>
<td>0.5</td>
<td>5.1278</td>
<td>3.1816</td>
</tr>
<tr>
<td>1.0</td>
<td>5.0359</td>
<td>3.1725</td>
</tr>
<tr>
<td>1.5</td>
<td>5.0678</td>
<td>3.1789</td>
</tr>
<tr>
<td>MSE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>–</td>
<td>5.0420</td>
<td>3.1767</td>
</tr>
</tbody>
</table>

4.3.2 Maximum Entropy Mapping

Optimal information transfer from the input distribution to the weights happen when the magnification factor is 1. Such a mapping should ensure that every node is active with equal probability, also called as equiprobabilistic mapping (Van Hulle, 2000). Since, the entropy is a measure that determines the amount of information content, we use Shannon’s entropy (4–18) of the weights to quantify this property.

$$l = -\sum_{r=1}^{M} \rho(r) \ln(\rho(r))$$  \hspace{1cm} (4–18)

where $\rho(r)$ is the probability of the node $r$ to be the winner.

Table 4-1 shows the change in $l$ with the change in the value of the kernel bandwidth and the performance between the SOM-MSE and the SOM-CIM is compared. The mapping here is generated in the batch mode with the input having 5000 2-dimensional random samples generated from a linear distribution $P(x) \propto 5x$ and mapped on to a 5x5 rectangular grid with $\sigma = 10 \rightarrow 0.0045$. 

32
Figure 4-4. The figures shows the input scatter plot and the mapping of the weights at the final convergence. (In case of SOM-CIM, the Gaussian kernel with $\sigma = 0.1$ is used.) The 'dots' indicate the weight vectors and the lines indicate the connection between the nodes in the lattice space.

From table 4-1, it is clear that as the value of the $\sigma$ influences the quality of the mapping and figure 4-4 shows the weight distribution at the convergence. It is clear that, for large $\sigma$ because of wider L2-norm region the mapping of the SOM-CIM behaves similar to that of the SOM-MSE. But as $\sigma$ is decreased the nodes try to be equiprobabiliteic indicating that they are not over sampling the low probability regions. But setting the value of $\sigma$ too small again distorts the mapping. This is because of the restricted movement of the nodes; which leads to over sampling the high probability region. This underlines the importance the value of $\sigma$ to obtain a good quality mapping.
Figure 4-5. Negative log likelihood of the input versus the kernel bandwidth. The figure shows how the negative log likelihood of the inputs given the weights change with the change in the kernel bandwidth. Note that the negative value in the plot indicates that the model, the weights and the kernel bandwidth, is not appropriate to estimate the input distribution.

It can also be observed how the type of the kernel also influences the mapping. The Gaussian and the Cauchy kernels give the best mapping at different values of $\sigma$, indicating that the shape of the L2 norm region for these two kernels are different.

Moreover, since it is shown that the SOM-CIM is equivalent to density estimation, the negative log likelihood of the input given the weights is also observed when the Gaussian kernel is used. The negative log likelihood is given by

$$LL = -\frac{1}{N}\sum_{n}^{N} \log(p(v(n)/W, \sigma))$$

(4–19)

where

$$p(v(n)/W, \sigma) = \frac{1}{M}\sum_{i}^{M} G_{\sigma}(v(n), w_{i})$$

(4–20)

. The change in its value with the change in the kernel bandwidth is plotted in the figure 4-5. For an appropriate value of the kernel bandwidth, the likelihood of the estimating the input is high; whereas for large values of $\sigma$ it decreases considerably.
CHAPTER 5
ADAPTIVE KERNEL SELF ORGANIZING MAPS

Now that the influence of the kernel bandwidth on the mapping is studied, one should understand the difficulty involved in setting the value of \( \sigma \). Though it is understood from the theory of Correntropy (Liu et al., 2007) that the value of \( \sigma \) should be usually less than the variance of the input data, it is not clear how to set this value. Since the SOM-CIM can be considered as a density estimation procedure, certain rules of thumb, like the Silverman's rule (Silverman, 1986) can be used. But it is observed that these results are suboptimal in many cases because of the Gaussian distribution assumption of the underlying data. It should also be noted that for equiprobabilistic modeling each node should adapt to the density of the inputs in it own vicinity and hence, should have its own unique bandwidth. In the following section we try to address this using an information theoretic divergence measure.

5.1 The Algorithm

Clustering of the data is closely related to the density estimation. An optimal clustering should ensure that the input density can be estimated using the weight vectors. Several density estimation based clustering algorithms are proposed using information theoretic quantities like divergence, entropy, etc. There are also several clustering algorithms in the ITL literature like the Information Theoretic Vector Quantization (Lehn-Schioler et al., 2005), Vector Quantization using KL-Divergence (Hegde et al., 2004) and Principle of Relevant Information (Principe, 2010); where quantities like KL-Divergence, Cauchy-Schwartz Divergence and Entropy are estimated non-parametrically. As we have shown earlier, the SOM-CIM with the Gaussian kernel can also be considered as a density estimation procedure and hence, a density estimation based clustering algorithm.

In each of these cases, the value of the kernel bandwidth plays an important role is the density estimation and should be set such that the divergence between the
true and the estimated pdf is as small as possible. Here we use the idea proposed by Erdogmus et al. (2006) of using the Kullback-Leibler divergence $D_{KL}$ to estimate the kernel bandwidth for density estimation and extend it to the clustering algorithms.

As shown in section 3.1.2, the $D_{KL}(f \parallel g)$ can be estimated non-parametrically from the data using the Parzen window estimation. Here if $f$ is the estimated pdf using the data and $g$ is the estimated pdf using the weight vectors, then the $D_{KL}$ can be written as

$$D_{KL}(f, g) = E_v \left[ \log \left( \sum_i^{N} G_{\sigma_v}(v - v(i)) \right) \right] - E_v \left[ \log \left( \sum_j^{M} G_{\sigma_w}(v - w(j)) \right) \right]$$  \hspace{1cm} (5–1)

where $\sigma_v$ is the kernel bandwidth for estimating the pdf using the input data and $\sigma_w$ is the kernel bandwidth while using the weight vectors to estimate the pdf. Since we want to adapt the kernel bandwidth of the weight vector, minimizing $D_{KL}$ is equivalent to minimizing the second term in 5–1. Hence, the cost function for adapting the kernel bandwidth is

$$J(\sigma) = -E_v \left[ \log \left( \sum_j^{M} G_{\sigma_w}(v - w(j)) \right) \right]$$ \hspace{1cm} (5–2)

This also called as the Shannon's cross entropy.

Now, the estimation of the pdf using the weight vector can be done using a single kernel bandwidth, homoscedastic, or with varying kernel bandwidth, heteroscedastic. We will consider each of these cases separately here.

5.1.1 Homoscedastic Estimation

If the pdf is estimated using the weights with a single kernel bandwidth, given by

$$g = \sum_j^{M} G_{\sigma_w}(v - w(j))$$ \hspace{1cm} (5–3)

then the estimation is called homoscedastic. In such a case only a single parameter needs to be adapted in 5–2. To adapt this parameter, gradient descent is performed over the cost function by taking the derivative of $J(\sigma)$ w.r.t $\sigma_w$. The gradient comes out to be

36
\[
\frac{\partial J}{\partial \sigma_w} = -E \left\{ \sum_j^M G_{\sigma_w}(v - w(j)) \left[ \frac{\|v - w(j)\|^2}{\sigma^2_{\sigma_w}} - \frac{d}{\sigma^2_{\sigma_w}} \right] \right\}
\]

To find an on-line version of the update rule, we use the stochastic gradient of the above update rule by replacing the expected value by the instantaneous value of the input. The final update rule comes out to be

\[
\triangle \sigma_w = -\left\{ \sum_j^M G_{\sigma_w}(v(n) - w(j)) \left[ \frac{\|v(n) - w(j)\|^2}{\sigma^2_{\sigma_w}} - \frac{d}{\sigma^2_{\sigma_w}} \right] \right\}
\]

\[
\sigma_w(n + 1) = \sigma_w(n) - \eta \triangle \sigma_w
\]

where \( d = \) input dimensions and \( \eta \) is the learning rate.

This can be easily extended to the batch mode by taking the sample estimation of the expected value and finding the stationary point of the gradient. It comes out to a fixed point update rule, indicating that it requires a few epochs to reach a minima. The batch update rule is

\[
\sigma^2_w = \frac{1}{Nd} \sum_n^N \sum_j^M G_{\sigma_w}(v(n) - w(j)) \left[ \frac{\|v(n) - w(j)\|^2}{\sigma^2_{\sigma_w}} \right] \sum_j^M G_{\sigma_w}(v(n) - w(j))
\]

It is clear from both on-line update rule 5–6 and batch update rule 5–7 that the value of \( \sigma_w \) is dependent on the variance of the error between the input and the weights localized by the Gaussian function. This localized variance converges such that the divergence between the estimated pdf and the true pdf of the underlying data is minimized. Hence, these kernels can be used to approximate the pdf density using the mixture of the Gaussian kernels centered at the weight vectors. We will get back to this density estimation technique in the later chapters.

The performance of this algorithm is shown in the following experiment. Figure 5-1 shows the adaptation process of the \( \sigma \) where the input is a synthetic data consisting of 4 clusters centered at four points on a semicircle distorted with the Gaussian noise of 0.4 variance. The kernel bandwidth is adapted using the on-line update rule 5–6 by...
keeping the weight vectors constant at the centers of the clusters. It is observed that the kernel bandwidth, $\sigma$, convergence to 0.4041. This nearly equal to the local variance of the inputs, i.e., the variance of the Gaussian noise at each cluster center.

However, the problem with the homoscedastic model is that if the window width is fixed across the entire sample, there is a tendency for spurious noise to appear in the tails of the estimates; if the estimates are smoothed sufficiently to deal with this, then essential detail in the main part of the distribution is masked. In the following section we try to address this by considering heteroscedastic model, where each node has unique a kernel bandwidth.

### 5.1.2 Heteroscedastic Estimation

If the density is estimated using the weight vectors with varying kernel bandwidth, given by

$$g = \sum_{j}^{M} G_{\sigma_{w}(j)}(v - w(j))$$

then the estimation is called heteroscedastic. In this case, the $\sigma$ associated with each node should be adapted to the variance of the underlying input density surrounding
Figure 5-2. Kernel adaptation in heteroscedastic case. [A] The plot shows the track of the $\sigma$ values associated with their respective weights against the number of epochs. Initially, the kernel bandwidths are chosen randomly between (0,1] and are adapted using the on-line mode. [B] The scatter plot of the weights and the inputs with the rings indicating the kernel bandwidth associated with each node.

The update rules come out to be:

**On-line Update rule:**

\[
\begin{align*}
\Delta \sigma_w(n, i) &= -\left\{ \frac{G_{\sigma_w(i)}(v(n) - w(i))}{\sum_{j=1}^{M} G_{\sigma_w(j)}(v - w(j))} \left[ \frac{||v-w(i)||^2}{\sigma_w(i)^3} - \frac{d}{\sigma_w(i)} \right] \right\} \\
\sigma_w(n + 1, i) &= \sigma_w(n, i) - \eta \Delta \sigma_w(n, i) 
\end{align*}
\] (5–10)

**Batch Mode Update rule:**

\[
\begin{align*}
\frac{\partial J}{\partial \sigma_w(i)} &= -E_v \left\{ \frac{G_{\sigma_w} (v - w(i)) \left[ \frac{||v-w(i)||^2}{\sigma_w(i)^3} - \frac{d}{\sigma_w(i)} \right]}{\sum_{j=1}^{M} G_{\sigma_w(j)} (v - w(j))} \right\} 
\end{align*}
\] (5–9)

Similar to the previous case, for on-line update rule we find the stochastic gradient and for batch mode update rule we replace the expected value with the sample estimation before finding the stationary point of the gradient.

The update rules come out to be:

**On-line Update rule:**

\[
\begin{align*}
\Delta \sigma_w(n, i) &= -\left\{ \frac{G_{\sigma_w(i)}(v(n) - w(i))}{\sum_{j=1}^{M} G_{\sigma_w(j)}(v - w(j))} \left[ \frac{||v(n)-w(i)||^2}{\sigma_w(i)^3} - \frac{d}{\sigma_w(i)} \right] \right\} \\
\sigma_w(n + 1, i) &= \sigma_w(n, i) - \eta \Delta \sigma_w(n, i) 
\end{align*}
\] (5–10)

**Batch Mode Update rule:**
\[ \sigma_w(i)^2 = \frac{1}{d} \sum_n^N \frac{\sum^m_{j=1} G_{\sigma_w(i)}(v(n) - w(j)) \left[ \|v(n) - w(i)\|^2 \right]}{\sum^m_{j=1} G_{\sigma_w(i)}(v(n) - w(j))} \] (5–12)

Figure 5-2 shows the kernel bandwidth adaptation for a similar setup as described for the homoscedastic case (5.1.1). Initially, the kernel bandwidths are considered randomly between (0,1] and are adapted using the on-line learning rule (5–11) while keeping the weight vectors constant at the centers of the clusters. The kernel bandwidth tracks associated with each node are shown and the \( \sigma \) associated with each node again converged to the local variance of the data.

### 5.2 SOM-CIM with Adaptive Kernels

It is easy to incorporate the above mentioned two adaptive kernel algorithms in the SOM-CIM. Since the value of the kernel bandwidth adapts to the input density local to the position of the weights, the weights are first updated followed by the kernel bandwidth. This ensures that the kernel bandwidth adapts to the present position of the weights and hence, better density estimation.

The adaptive kernel SOM-CIM in case of homoscedastic components can be summarized as:

- The winner is selected using the local error as
  \[
  r = \arg \min_s \sum_t^M h_{st} \left( G_{\sigma(n)}(0) - G_{\sigma(n)}(\|v(n) - w_t\|) \right) \] (5–13)

- The weights and then the kernel bandwidth are updated as
  \[
  \Delta w_s = -h_{is} G_{\sigma(n)}(\|v(n) - w_s\|) \frac{(v(n) - w_s)}{\sigma(n)^3} \\
  w_s(n + 1) = w_s(n) - \eta \Delta w_s \] (5–14)

  \[
  \Delta \sigma(n) = -\left\{ \frac{\sum_j^M G_{\sigma(n)}(v(n) - w_j(n + 1) \left[ \|v(n) - w_j(n + 1)\|^2 \right]}{\sigma(n)^3} - \frac{d}{\sigma(n)} \right\} \\
  \sigma(n + 1) = \sigma(n) - \eta_{\sigma} \Delta \sigma(n) \] (5–16)
• In the batch mode, the weights and the kernel bandwidth are updated as

\[
\mathbf{w}^+_s = \frac{\sum_n h_{rs} G_{sr}(\mathbf{v}(n) - \mathbf{w}_s) \mathbf{v}(n)}{\sum_n h_{rs} G_{sr}(\mathbf{v}(n) - \mathbf{w}_s)} \tag{5-18}
\]

\[
\sigma^+ = \frac{1}{Nd} \sum_n \sum_j^M G_{sr}(\mathbf{v}(n) - \mathbf{w}_j) \left[ \frac{||\mathbf{v}(n) - \mathbf{w}_j||^2}{\sum_j^M G_{sr}(\mathbf{v}(n) - \mathbf{w}_j)} \right] \tag{5-19}
\]

In case of heteroscedastic kernels, the same update rules apply for the weights but with \(\sigma\) specified for each node; while the kernel bandwidths are updated as

**On-line mode:**

\[
\Delta \sigma_i(n) = \left\{ \frac{G_{sr}(\mathbf{v}(n) - \mathbf{w}_i(n + 1)) \left[ \frac{||\mathbf{v}(n) - \mathbf{w}_i(n + 1)||^2}{\sigma_i(n)} \right]}{\sum_j^M G_{sr}(\mathbf{v}(n) - \mathbf{w}_j(n + 1))} \right\} \tag{5-20}
\]

\[
\sigma_i(n + 1) = \sigma_i(n) - \eta \Delta \sigma_i(n) \tag{5-21}
\]

**Batch mode:**

\[
\sigma_i^+ = \frac{1}{d} \sum_n \sum_j^M \frac{G_{sr}(\mathbf{v}(n) - \mathbf{w}_j)(||\mathbf{v}(n) - \mathbf{w}_j||^2)}{G_{sr}(\mathbf{v}(n) - \mathbf{w}_j)||\mathbf{v}(n) - \mathbf{w}_j||} \tag{5-22}
\]

But it is observed that when the input has a high density part, compared to the rest of the distribution, the kernel bandwidth of the weights representing these parts shrinks too small to give any sensible similarity measure and makes the system unstable. To counter this, a scaling factor \(\rho\) can be introduced as

\[
\Delta \sigma_i(n) = \left\{ \frac{G_{sr}(\mathbf{v}(n) - \mathbf{w}_i(n + 1)) \left[ \frac{||\mathbf{v}(n) - \mathbf{w}_i(n + 1)||^2}{\sigma_i(n)} \right]}{\sum_j^M G_{sr}(\mathbf{v}(n) - \mathbf{w}_j(n + 1))} \right\} \tag{5-23}
\]

This ensures that the value of the \(\sigma\) does not go too low. But at the same time it inadvertently increases the kernel bandwidth of the rest of the nodes and might result in a suboptimal solution. The experiment similar to that in section 4.3.1 clearly demonstrates this. 100,000 samples from the distribution \(f(x) = 2x\) are mapped on to a 1D chain of 50 nodes in on-line mode and the relation between the weights and the node indexes is observed. Figure 5-3 shows the results obtained. Clearly in case of the homoscedastic case the mapping converge to the ideal mapping. But in case of
Figure 5-3. Magnification of the map in homoscedastic and heteroscedastic cases. The Figures [A] and [B] shows the tracks of the kernel bandwidth in case of homoscedastic and heteroscedastic kernels, respectively. Figures [C] and [D] shows the plot between $\ln(r)$ and $\ln(w)$. '-' indicates the ideal plot representing the relation between $w$ and $r$ when the magnification is 1 and '...' indicates the ideal plot representing the magnification of 2/3. '-+-' indicates relation obtained with adaptive kernels SOM-CIM in homoscedastic and heteroscedastic cases, respectively. Refer text for explanation.

In the heteroscedastic kernels it is observed that the system is unstable for $\rho = 1$. Here to obtain a stable result $\rho$ is set to 0.7. This stabilizes the mapping by not allowing the kernel bandwidth to shrink too small. But it clearly increases the kernel bandwidth of the nodes in low probability region and hence, the distortion in 5-3D.

It is also interesting to see the changes in the kernel bandwidth while adapting in the Figure 5-3A and Figure 5-3B. Initially, all the kernels converge to the same
Table 5-1. The information content and the mean square quantization error for homoscedastic and heteroscedastic cases in SOM-CIM. $I_{\text{max}} = 3.2189$

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean Square Quantization Error</th>
<th>Entropy or Info Cont., I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homoscedastic kernels</td>
<td>$5.8725 \times 10^{-3}$</td>
<td>3.2095</td>
</tr>
<tr>
<td>Heteroscedastic kernels ($\rho = 0.5$)</td>
<td>$6.1179 \times 10^{-3}$</td>
<td>3.1970</td>
</tr>
<tr>
<td>Heteroscedastic kernels ($\rho = 1$)</td>
<td>$15.268 \times 10^{-3}$</td>
<td>2.700</td>
</tr>
<tr>
<td>Constant kernel $^1$, $\sigma = 0.1$</td>
<td>$5.7240 \times 10^{-3}$</td>
<td>3.2101</td>
</tr>
<tr>
<td>MSE</td>
<td>$5.0420 \times 10^{-3}$</td>
<td>3.1767</td>
</tr>
</tbody>
</table>

Figure 5-4. Mapping of the SOM-CIM with heteroscedastic kernels and $\rho = 1$.

bandwidth as the nodes are concentrated at the center and as map changes from the ordering phase to the convergence the kernel bandwidths adapt to the local variance of each node. This kind of adaptation ensures that the problem of slow annealing of the neighborhood function, discussed in section 4.1, can be resolved by having relatively large bandwidth at the beginning, allowing the free movement of the nodes during ordering phase.

As the adaptive kernel algorithm is able to map the nodes near to the optimal mapping, it is expected to transfer more information about the input density to the

$^1$ The best result from Table 4-1 is reproduced for comparison.
weights. Results obtained for the experiment similar to the one in section 4.3.2, where the inputs are 5000 samples from the 2 dimensional distribution with pdf $f(x) = 5x$ mapped on to a 5x5 rectangular grid, are shown in Table 5-1. With the homoscedastic kernels, the kernel bandwidth converged to 0.0924, which is close to the value that is obtained as the best result in Table 4-1 ($\sigma = 0.1$) and at the same time is able to transfer more information to the weights. But in the heteroscedastic case the system is unstable with $\rho = 1$ and fails to unfold (5-4), concentrating more on the high probability regions. On the other hand, when $\rho$ is set to 0.5, the resulting map does not provide a good output because of the large $\sigma$ values. Figure 5-5 clearly demonstrates this. When $\rho = 0.5$, it clearly over samples the low probability regions because of the large kernel bandwidths of the nodes representing them.
CHAPTER 6
APPLICATIONS

Now that we saw how the SOM can be adapted using the CIM, in this chapter we try to use the SOM-CIM in few of the applications of the SOM; like density estimation, clustering and principal curves, and show that how it can improve the performance.

6.1 Density Estimation

Clustering and density estimation are closely related and one is often used to find the other, i.e., density estimation is used to find the clusters (Azzalini and Torelli, 2007) and clustering is used to find the density estimation (Van Hulle, 2000). As we have shown earlier SOM-CIM is equivalent to a density estimation procedure and with the adaptive kernels it should be able to effectively reproduce the input density using the Parzen non-parametric density estimation procedure (Parzen, 1962).

Here we try to estimate a 2 dimensional Laplacian density function shown in Figure 6-1A. The choice of the Laplacian distribution is appropriate to compare these methods because it is heavy tail distribution and hence, is difficult to estimate without the proper choice of the kernel bandwidth.

Figure 6-1 and Table 6-1 shows the results obtained when different methods are used to estimate the input density. Here 4000 samples of a 2 dimensional Laplacian distribution are mapped on to a 10x10 rectangular grid. The neighborhood function is given by 4–13 and is annealed using 4–14 with $\theta = 0.3$ and $\sigma_h = 20$. When the SOM-MSE is used the kernel bandwidth is estimated using Silverman’s rule (Silverman, 1986):

$$\sigma = 1.06\sigma_f N^{-\frac{1}{5}}$$  \hspace{1cm} (6–1)

where $\sigma_f$ is the variance of the input and $N$ is the number of input samples. Figure 6-1B clearly demonstrates that this procedure not able to able to produce a good result because of the large number of bumps in the low probability regions. This is because
Figure 6-1. Results of the Density estimation using SOM. [A] A 2 dimensional Laplacian density function. [B] The estimated density using SOM with MSE. The kernel bandwidth is obtained using the Silverman’s rule. [C] The estimated density using SOM-CIM with homoscedastic kernels. [D] The estimated density using SOM-CIM with heteroscedastic kernels.

of the oversampling of the low probability regions when MSE is used, as discussed in section 2.3.

On the other hand, the use of the SOM-CIM, with both homoscedastic and heteroscedastic kernels, is able to reduce the oversampling of the low probability regions. But in case of homoscedastic kernels, because of the constant bandwidth of the kernels for all the nodes, the estimation of the tail of the density is still noisy and is unable to clearly demonstrate the characteristics of the main lobe of the density properly.
Table 6-1. Comparison between various methods as density estimators. MSE and KLD are the mean square error and KL-Divergence, respectively, between the true and the estimated densities.

<table>
<thead>
<tr>
<th>Method</th>
<th>MSE ($\times 10^{-4}$)</th>
<th>KLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOM with MSE</td>
<td>2.0854</td>
<td>308.4163</td>
</tr>
<tr>
<td>SOM-CIM, homoscedastic</td>
<td>2.0189</td>
<td>128.6758</td>
</tr>
<tr>
<td>SOM-CIM, heteroscedastic ($\rho = 0.8$)</td>
<td>3.1432</td>
<td>12.6948</td>
</tr>
</tbody>
</table>

This is resolved when the heteroscedastic kernels are used. Because of the varying kernel bandwidth, it is able to smooth out the tail of the density while still retaining the characteristics of the main lobe and hence, able to reduce the divergence between the true and estimated densities.

### 6.2 Principal Surfaces and Clustering

#### 6.2.1 Principal Surfaces

Topology preservation maps can be interpreted as an approximation procedure for the computation of principal curves, surfaces or higher-dimensional principal manifolds (Ritter et al., 1992). The approximation consists in the discretization of the function $f$ defining the manifold. The discretization is implemented by means of a lattice $A$ of corresponding dimension, where each weight vector indicates the position of a surface point in the embedding space $V$. Intuitively, these surface points, and hence the principal curve, are expected to pass right through the middle of their defining density distribution. This definition of principal surfaces in 1 dimensional case, and can be generalized to high dimensional principal manifold as (Ritter et al., 1992):

Let $f(s)$ be a surface in the vector space $V$, i.e, $\dim(f) = \dim(V) - 1$, and let $d_r(v)$ be the shortest distance of a point $v \in V$ to the surface $f$. $f$ is a principal surface corresponding to the the density distribution $P(v)$ in $V$, if the “mean squared distance”

$$D_f = \int d_r^2(v) P(v) d^d v$$

is extremal with respect to local variation of the surface.
Figure 6-2. Clustering of the two-crescent dataset in the presence of outlier noise. [A] The mapping at the convergence of SOM with MSE. [B] - [D] The scatter plot of the weights with the kernel bandwidth at different epochs during learning.

But the use of the MSE as the criteria for the goodness of the principal surfaces, makes it weakly defined because only the second order moments are used and also the distortion in the mapping when outliers are present. Figure 6-2A shows this when the two crescent data is slightly distorted by introducing some outliers. On the other hand, if the principal surface is adapted in the correntropy induced metric sense, then the effect of the outliers can be eliminated and gives a better approximation of the principal surfaces. Figure 6-2 illustrates this in case of the SOM-CIM with homoscedastic kernels,
Figure 6-3. The scatter plot of the weights showing the dead units when CIM and MSE are used to map the SOM. [A] The boxed dots indicate the dead units mapped when MSE is used. [B] Mapping when CIM with $\sigma = 0.5$ is used in the SOM. There are no dead units in this case.

where the kernel bandwidth adapts such that the outliers does not have significant effect the final mapping throughout the learning.

6.2.2 Avoiding Dead Units

Another problem with the SOM is that it can yield nodes that are never active, called as dead units. These units will not sufficiently contribute to the minimization of the overall distortion of the map and, hence, this will result in a less optimal usage of the map’s resources (Van Hulle, 2000). This is acute when there are clusters of data that are far apart in the input space.

The presence of the dead units can also be attributed to the MSE based cost function which pushes the nodes into these regions. Figure 6-3A indicates this where the input contains three nodes, each is skewed by 500 samples of a 2 dimensional Gaussian noise with variance equal to 0.25. On the other hand, when CIM is used, as indicated in section 4.3.2, the mapping tries to be equiprobabilistic and hence, avoids dead units (Figure 6-3B).
Table 6-2. Number of dead units yielded for different datasets when MSE and CIM are used for mapping. Each entry is an average over 30 runs.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Dead Units</th>
<th>MSE (10^{-2})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artificial</td>
<td>MSE</td>
<td>4</td>
<td>1.3877</td>
</tr>
<tr>
<td></td>
<td>CIM, $\sigma = 0.2147$</td>
<td>0</td>
<td>8.4458</td>
</tr>
<tr>
<td>Iris</td>
<td>MSE</td>
<td>5.6</td>
<td>5.939</td>
</tr>
<tr>
<td></td>
<td>CIM, $\sigma = 0.5$</td>
<td>1</td>
<td>8.023</td>
</tr>
<tr>
<td>Blood Transfusion</td>
<td>MSE</td>
<td>0</td>
<td>0.7168</td>
</tr>
<tr>
<td></td>
<td>CIM, $\sigma = 0.2$</td>
<td>0</td>
<td>0.8163</td>
</tr>
</tbody>
</table>

Table 6-2 show the results obtained over three different datasets mapped on to 10x5 hexagonal grid. The artificial dataset is the same as the one described above. The Iris and Blood transfusion datasets are obtained from the UC Irvine repository.

The Iris dataset contains 3 classes of 50 instances each, where each class refers to a type of iris plant. Here, one class is linearly separable from the other 2; the latter are not linearly separable from each other, and each instance has 4 features. It is observed that the dead units appear in between the two linearly separable cases. The blood transfusion dataset (normalized to be between [0,1] before mapping) contains 2 classes with 24% positive and 76% negative instances. Though there are no dead units in this case, as we will show later, this kind of mapping neglects the outliers and hence able to give a better visualization of the data.

An other important observation is, the adaptive kernel algorithms are unable to give a good result for both Iris and Blood Transfusion datasets. The reason for this is the inherent nature of the Parzen density estimation, which is the center piece for this algorithm. As the number of dimensions increase the number of input samples required by the Parzen density estimation for proper density estimation increases exponentially. Because of the limited amount of data in these datasets the algorithm failed to adapt the kernel bandwidth.
Figure 6-4. The U-matrix obtained for the artificial dataset that contains three clusters. [A] U-matrix obtained when the SOM is trained with MSE and the euclidian distance is used to create the matrix. [B] U-matrix obtained when the CIM is used to train the SOM and to create the matrix.

6.3 Data Visualization

Visualizing high dimensional data in low dimensional grid is one of the important applications of SOM. Since the topography can be preserved on to the grid, visualizing the grid can give a rough idea about the structure of the data distribution and, in turn, an idea about the number of clusters without any apriori information. The popular SOM visualization tool is the U-matrix, where the distance between the a node and all its neighbors is represented in a gray scale image. If the euclidean distance is used to construct the matrix, then the distance between the closely spaced clusters is over shadowed by the far away clusters, making it difficult to detect them. On the other hand, because of the non-linear nature of the CIM and saturation region beyond certain point (determined by the value of $\sigma$), the closely spaced clusters can be detected if the U-matrix is constructed using CIM.

Figure 6-4 shows the results obtained when the MSE and the CIM are used to train the SOM and to construct the U-matrix. The SOM here is a 10x5 hexagonal grid and
Figure 6-5. The U-matrix obtained for the Iris and blood transfusion datasets. [A] and [C] show the U-matrix for the Iris and Blood transfusion datasets, respectively, when the SOM is trained with MSE and the euclidian distance is used to create the matrix. [B] and [D] show their corresponding U-matrix obtained when the CIM is used to train the SOM and to create the matrix.
the input is a 2 dimensional artificial dataset containing 3 clusters; one cluster is farther away from the other two (6-3), each made up of 500 instances of the Gaussian noise with variance 0.25. Clearly, the boundaries are not crisp when the MSE is used. This is because of the dead units the SOM yields when the MSE is used. On the other hand, the U-matrix obtained when the SOM-CIM is used is able to give a crisp boundaries.

The problem of viewing closely space clusters when there are other outliers or faraway clusters becomes even more difficult. Figure 6-5 illustrates this is the case of the Iris dataset and the Blood transfusion dataset. The Iris dataset has three clusters where only one cluster is linearly separable from the other two. When the U-matrix is created using the MSE, only the linearly separable cluster is distinguishable from the other two. Where as, the U-matrix created using the CIM is able to show the three clusters (though not with crisp boundaries). The reason for this is not only because of avoiding the dead units but also because of the non-linear norm of the CIM. If the distance between the nodes that are closely space (in euclidian sense) is evaluated using the CIM, then this is not over shadowed by the distance evaluated using the CIM between the nodes that are far apart because of the saturation of the CIM norm.

The blood transfusion dataset has two clusters and expected to have large number of outliers. The quality of the U-matrix is not discussed and is left to the readers.
The use of the kernel Hilbert spaces by Vapnik (1995) has spurred the use of the kernel methods in several fields. This idea is also used in the self-organizing maps previously by Lau et al. (2006), Andras (2002), MacDonald (2000), etc. In all these cases, the use of the kernel trick is viewed in the context of pattern classification and the increase in performance is attributed to the assumption that the classes tend to be linearly separable when mapped into higher dimensions.

On the other hand, the use of the correntropy induced metric gives an idea about how the kernelization affects in the input space and hence, is able to give an idea about how the final output of the mapping is affected by the choice of the kernel and kernel bandwidth. As indicated, the choice of the kernel bandwidth dictates the magnification of the mapping. For example, larger bandwidth adapts the SOM-CIM to minimize the quantization error because of the large L2-norm induced, whereas smaller bandwidth might result in a map that concentrates more on the high probability parts; and hence, distorting the mapping. Also, as previously discussed, the advantage of using the CIM lies in its strong outlier’s rejection capability and the presence of higher order moments. Both these properties are useful when the input distribution is non-uniform and hence, the SOM-CIM can outperform the SOM only when the data is non-uniformly distributed (true for many practical cases).

The proposed adaptive kernel algorithm based on the KL-divergence is able to adapt the bandwidth nearly to the optimal solution. But it is observed that the algorithm is unstable in the heteroscedastic case and an additional free parameter $\rho$ needs to be specified. In spite of that, the final mapping is still less sensitive to the value of $\rho$ than the value of $\sigma$ and is also able to give a heteroscedastic alternative.

Another point that needs to be discussed is the extension of the use of the CIM in other clustering algorithms like neural-gas, elastic nets, soft-topographic vector...
quantization\textsuperscript{1}, etc. The similarity measure used in the ranking of the weight vectors in the neural gas algorithm can be replaced by the CIM and a similar procedure can be applied to adapt the network. This is also expected to improve the performance of the network in terms of the magnification factor.

Finally, though the dependence of the magnification factor on the kernel bandwidth is shown experimentally, the analytical analysis is still illusive. The future work should concentrate on the finding the relation between the magnification factor and the kernel bandwidth; which in turn depends on the variance of the data. Also the use of multivariate kernels might be important when the SOM-CIM is used for density estimation and it also needs to be studied. The proposed adaptive kernel algorithm can also be extended to the multivariate kernels using the idea proposed by Erdogmus et al. (2006) but will lead to a larger computational complexity. An effective, less computationally expensive algorithm to adapt the kernel bandwidth is necessary.

\textsuperscript{1} This is discussed in Appendix B
APPENDIX A
MAGNIFICATION FACTOR OF THE SOM WITH MSE

The quantitative analysis of the magnification factor of the SOM with the MSE as the cost function is shown here. Detailed analysis of this can be found at (Ritter et al., 1992).

The along as the neighborhood function $h_{ps}$ is non-zero, $w(r, t)$ undergoes a non-zero change at each time step. This transition of the weights from $w$ to $\hat{w}$ can be formally by the transformation

$$\mathbf{w} = T(\hat{\mathbf{w}}, \mathbf{v}, \epsilon)$$

where $\epsilon$ is the learning rate. Explicitly, the transition in A–1 can be written as

$$\mathbf{w}_s = \hat{\mathbf{w}}_s + \epsilon h_{ps}(\mathbf{v} - \hat{\mathbf{w}}_s), \quad \forall s$$

Since the input $\mathbf{v}$ occurs only with the probability $P(\mathbf{v})$, the probability $Q(\mathbf{w}, \hat{\mathbf{w}})$ of the transition from the state $\mathbf{w}$ to $\hat{\mathbf{w}}$ can be written as

$$Q(\mathbf{w}, \hat{\mathbf{w}}) = \int \delta(\mathbf{w} - T(\hat{\mathbf{w}}, \mathbf{v}, \epsilon))P(\mathbf{v})d\mathbf{v}$$

In the equilibrium condition, the necessary and sufficient condition is that the transition probability to be zero. This implies

$$Q(\mathbf{w}, \hat{\mathbf{w}}) = 0$$

$$\Rightarrow \int \epsilon h_{ps}(\mathbf{v} - \mathbf{w}_s)P(\mathbf{v})d\mathbf{v} = 0 \forall s$$

A maximally ordered map, defined as a map that is fully unfolded without any 'twists' or 'kinks', is assumed which leads to the validity of the following assumptions in the continuum case

- It is assumed that for a sufficiently large systems $\mathbf{w}$, is a function of that varies slowly from lattice point to lattice point so that its replacement by a function $w(r)$ on a continuum of $r$-values is justified.
- We assume that $w(r)$ is one-to-one.
Also the neighborhood function with a steep maximum at \( r = s \) that satisfies

\[
\int h_s(r - s) dr = 0
\]
\[
\int h(r - s)(r_i - s_i)(r_j - s_j) dr = \delta_{ij} \sigma_n^2, \quad \forall i, j = 1, 2, \ldots, d
\]  

(A–5)

where \( d \) is the dimensionality of the lattice, which is assumed same as the dimensionality of the input space \( V \) and \( \sigma_n \) is the neighborhood range of the Gaussian neighborhood function used.

Due to the continuum approximation where it is assumed that there are infinite number of nodes, at the equilibrium, the quantity \( \| v - w_r \| \) vanishes since each sample \( v \) in the input space has a corresponding to an \( r \) for which \( v = w_r \). So, we can replace \( v \) in A–4 with \( w_r \). Also, by changing the variable of integration to \( s \) instead of \( w \), the A–4 becomes

\[
\int h_{is}(w_i - w_s) P(w_i) J(w_i) ds = 0
\]  

(A–6)

Here

\[
J(w_r) = \left| \frac{dv}{s} \right|
\]  

(A–7)

is the absolute value of the Jacobian of the map \( \phi_w : v \rightarrow w \). With \( q = r - s \) as a new integration variable and \( \tilde{P}(r) = P(w(r)) \) the expansion of A–6 in the powers of \( g \) yields (with implicit summation \( q_i \partial_i \) is to be summed over all \( i \))

\[
0 = \int h_{q} (q_i \partial_i w + \frac{1}{2} q_i q_j \partial_i \partial_j w + \ldots) . (\tilde{P} + \partial_k \tilde{P} + \ldots)(J + q_i \partial_i J + \ldots) dq
\]

\[
= \int h_{q} (q_i \partial_i w \tilde{P} J + q_i q_k \partial_i w \partial_k \tilde{P} J + q_i q_i \partial_i w \partial_i J \tilde{P} + \ldots)dq
\]

\[
= \int h_{q} dq . ([\partial_i w] \partial_i (\tilde{P} J) + \frac{1}{2} \tilde{P} J \partial_i \partial_i w) (s) + O(\sigma_n^4)
\]

\[
= \sigma_n^2 [\partial_i w] \partial_i (\tilde{P} J) + \frac{1}{2} \tilde{P} J \partial_i \partial_i w] (r) + O(\sigma_n^4)
\]  

(A–8)

(A–9)
where the condition about the neighborhood function in A–5 are used. The necessary and sufficient condition for A–9 to hold for small $\sigma$ is

$$\sum_i \partial_i \mathbf{w} \left( \frac{\partial_i \tilde{P}}{\tilde{P}} + \frac{\partial_i J}{J} \right) = -\frac{1}{2} \sum_i \partial_i^2 \mathbf{w}$$

(A–10)

Substituting the Jacobian matrix $J_i = \partial_i \mathbf{w}_i(r)$ and $\triangle = \sum_i \partial_i^2$, the condition becomes

$$\mathbf{J} \ln(\tilde{P}, \mathbf{J}) = -\frac{1}{2} \triangle \mathbf{w}$$

(A–11)

For a one-dimensional case we obtain $\mathbf{J} = J = \frac{d\mathbf{w}}{dr}$ and $\triangle \mathbf{w} = \frac{d^2 \mathbf{w}}{dr^2}$ with $\mathbf{w}$ and $r$ as scalars. Using this the differential equation A–10 can be solved. Rewriting the above equation by substituting these values, we get

$$\frac{d\mathbf{w}}{dr} \left( \frac{1}{\tilde{P}} \frac{d\tilde{P}}{dr} + \left( \frac{d\mathbf{w}}{dr} \right)^{-1} \frac{d^2 \mathbf{w}}{dr^2} \right) = -\frac{1}{2} \frac{d^2 \mathbf{w}}{dr^2}$$

(A–12)

from which we can obtain

$$\frac{d}{dr} \ln \tilde{P} = -\frac{3}{2} \frac{d}{dr} \ln \left( \frac{d\mathbf{w}}{dr} \right)$$

(A–13)

Since $\phi_{\mathbf{w}}(\mathbf{w}(r)) = r$ holds, the local magnification factor $M$ of $\phi_{\mathbf{w}}$ can be defined by $M = 1/J$. For the one-dimensional case $M = (\frac{d\mathbf{w}}{dr})^{-1}$ and we obtain as a relation between input stimulus distribution and the lattice representation

$$M(v) = J^{-1} = \frac{dr}{d\mathbf{w}} \propto P(v)^{2/3}$$

(A–14)

Hence, the local magnification factor $M(v)$ is dependent on the probability density $P(v)$ according to a power law with factor $2/3$.

The condition that need to be solved for obtaining the magnification factor of the SOM-CIM, at the equilibrium condition, is

$$\int h_{\text{is}} G_{\rho}(\mathbf{v} - \mathbf{w}_s)(\mathbf{w}_i - \mathbf{w}_s) P(\mathbf{v})d\mathbf{v} = 0$$

(A–15)

This is left for the future work.
Graepel et al. (1997) has used deterministic annealing (Rose, 1998) in topographic preserving vector quantization. The idea is to reduce the free energy of the clustering nodes while maintaining the topology of the nodes in the lattice space. The algorithm can be described in the following sections.

**B.1 SOM and Free Energy**

With the local error in deciding the winner, the quantization error over the entire dataset is given by

\[
F_{\text{quant}}(P, W) = \sum_{n} \sum_{r} p_{r}^{n} \sum_{s} h_{ss} D(v_{n}, w_{s})
\]  

(B–1)

where \( p_{r}^{n} \) is the posterior probability that the node \( r \) is the winner when \( v_{n} \) is considered and the entropy term which determines the probability of the nodes to occur is given by

\[
F_{\text{ent}}(P) = \sum_{n} \sum_{r} p_{r}^{n} \log \frac{p_{r}^{n}}{q_{r}}
\]  

(B–2)

where \( q_{r} \) = prior probability of the nodes. Using the B–1 and B–2, the free energy of the function can be written as

\[
F(P, W) = \beta F_{\text{quant}}(P, W) + F_{\text{ent}}(P)
\]  

(B–3)

where \( \beta \) is called the inverse temperature, larger the \( \beta \) lesser the influence of the entropy (Heskes, 2001).

**B.2 Expectation Maximization**

As proved by Neal and Hinton (1998) that the (local) minimum of the free energy corresponds to the (local) minimum error of the self organizing maps, expectation maximization can be used to find the optimal solution (Heskes, 2001; Neal and Hinton, 1998) of the weights and their corresponding posteriors.

The expectation step corresponds to finding the posteriors, \( p_{r}^{n} \), that minimizes the \( F(P, W) \). By using a Lagrange multiplier with the condition that \( \sum_{r} p_{r}^{n} = 1 \), the optimal
Figure B-1. The unfolding of the SSOM mapping when the CIM is used as the similarity measure. The snapshots are taken at intermediate point during the unfolding of the mapping at different values of $\beta$. The dataset considered here is a uniform distribution of 1000 points within a unit square. The neighborhood function is fixed at $\sigma_h = 0.5$.

Posteriors can be obtained and it comes out to be

$$p_t^n = \frac{q_t \exp(-\beta \sum_i h_{nt} D(v_{nt}, w_t))}{\sum_s q_s \exp(-\beta \sum_i h_{st} D(v_{nt}, w_t))} \quad (B-4)$$

This update rule does not depend on the distance measure and hence, would be same while using both the MSE and CIM measures.
Whereas the maximization step changes with the similarity measure.

**With MSE**

By finding the derivative of Eq. 5 w.r.t weights, \( w_s \), and equating it to zero, we get the updation rule as in Eq. 9.

\[
    w_s(i) = \frac{\sum_n \sum_r p_n^r h_{rs} x_n^r(i)}{\sum_n \sum_r p_n^r h_{rs}} \tag{B-5}
\]

**With CIM**

While using the CIM, the fixed point update rule is

\[
    w_s(i) = \frac{\sum_n \sum_r p_n^r h_{rs} G_\sigma(v_n - w_s) v_n^r(i)}{\sum_n \sum_r p_n^r h_{rs} G_\sigma(v_n - w_s)} \tag{B-6}
\]

With \( h_{rs} \rightarrow \delta \), STVQ acts as a traditional SOM with soft membership function (Graepel et al., 1997). Figure B-1 shows the evolution of the soft SOM (SSOM) when CIM with \( \sigma = 0.5 \) is used. Since the use of deterministic annealing is to avoid the local minima of the cost function, the discussion about the magnification of the mapping still holds here. It is expected that the use of the CIM here plays a similar role as in the case of the SOM-CIM.
REFERENCES


BIOGRAPHICAL SKETCH

Rakesh Chalasani is born in Mopidevi, India in 1986. He received his Bachelor of Technology degree from Department of Electronics and Communication Engineering at National Institute of Technology, Nagpur, India in 2008. He joined the Department of Electrical and Computer Engineering at the University of Florida in August 2008. He has been working with Dr. José C. Principe since 2009 and received his Master of Science degree in May 2010. His present research interests include machine learning, pattern recognition, unsupervised learning and adaptive signal processing.