Self Organizing Maps with Information Theoretic Learning

Rakesh Chalasani[1], Jose C. Principe

Computational NeuroEngineering Lab,
University of Florida, Gainesville, FL, USA - 32608

Abstract

The self organizing map (SOM) is one of the popular clustering and data visualization algorithms and has evolved as a useful tool in pattern recognition, data mining 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 oversampling the low probability regions.

In this work, we first discuss the kernel SOM in terms of a similarity measure called correntropy induced metric (CIM) and empirically show that this can enhance the magnification of the mapping without much increase in the computational complexity of the algorithm. We also show that adapting the SOM in the CIM sense is equivalent to reducing the localized cross information potential, an information-theoretic function that quantifies the similarity between two probability distributions. Using this property we propose a kernel bandwidth adaptation algorithm for Gaussian kernels, with both homoscedastic and heteroscedastic components. We show that the

*Corresponding author

proposed model can achieve a mapping with optimal magnification and can automatically adapt the parameters of the kernel function.
Keywords:
SOM, Kernel Methods, Information Theoretic Learning, Magnification factor

---

1. Introduction

The topographic organization of the sensory cortex is one of the complex phenomena in the brain and is widely studied. It is observed that neighboring neurons in the cortex are stimulated from neighboring stimuli in the input space and such organization of the neurons is called neighborhood preservation or topology preservation map. In other words, the cortical neural layer acts as a *topographic feature map*, where 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 cortex correspond to stimuli with similar features [24].

Inspired by such a biological plausibility, but not intending to explain it, Kohonen [14] 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, Kohonen's algorithm [15] is a neighborhood preserving vector quantization tool working on the winner-take-all principle, where the winner is determined as the most similar node to the input at an instant of time, also called 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 the grid of nodes in the output.

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 [24]. This magnification can be explicitly expressed as a power law between the input data density $P(\mathbf{v})$ and the weight vector density $P(\mathbf{w})$ at the time of convergence. The exponent is called as the *magnification factor or magnification exponent* [33]. A faithful representation of the data by the weights can happen only when this magnification factor is 1. However, it is shown that if 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* [24]. So, such a mapping is not able to transfer optimal information from the input data to the weight vectors.

The reason for the sub-optimal mapping of the traditional SOM algorithm can be attributed to the use of the Euclidean 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, which are the data in the low probability regions. This leads to oversampling of the low probability regions and undersampling the high probability regions by the weight vectors. On the other hand, kernel SOM [2] is shown to improve the performance in tasks like classification; however the reason for this improvement is not discussed.

Here, we first discuss the relationship between using the kernel "trick" and using a localized similarity measure called correntropy induced metric (CIM) to train the SOM. The goal of this work is to show that the proper-

ties of the CIM can help enhance the magnification of the SOM. Also, we show that the relation between KSOM and the other information-theoretic learning (ITL) [22] based measures can help to understand the use of the CIM in this context. Such insight can also help to adapt the free parameter in correntropy, the kernel bandwidth, leading to an optimal solution.

2. Back ground and previous work

2.1. Self organizing maps

Kohonen's original algorithm [14] of self organizing maps (SOM) is inspired by vector quantization, in which a group of inputs are quantized by a few weight vectors called nodes. However, in addition to the quantization of the inputs, here the nodes are arranged in a regular, low dimensional grid and the order of the grid is maintained throughout learning. Hence, the distribution of the input data in the high dimensional space can be preserved on the low-dimensional grid [15].

However, Erwin et al. [9] have shown that in the case of a finite set of training patterns the energy function of the SOM is highly discontinuous and in the case of continuous inputs the energy function does not exist. It is clear that things go wrong at the edges of the Voronoi regions where the input is equally close to two nodes. To overcome this, Heskes [13] has proposed that with a slight variation in the selection of the BMU, there can be a well defined energy function for the SOM. We briefly describe this here.

Before going further into the details of the algorithm, please note that the following notation is used throughout this work: The input distribution $\mathbf{V} \subset \mathbb{R}^d$ is mapped by the function Φ: $\mathbf{V} \rightarrow \mathbf{A}$, where $\mathbf{A}$ is in a lattice of M neurons, with each neuron having a weight vector $\mathbf{w_i} \in \mathbb{R}^d$, where $\mathbf{i}$ are

lattice indices.

Now, the learning algorithm for the SOM can be described as follows:

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

$$\mathbf{r} \quad = \quad \arg\min_{s} \sum_{t} h_{\mathbf{ts}} D(\mathbf{v}(n) - \mathbf{w_t}) \tag{1}$$

  where node $\mathbf{r}$ is the index of "winning" node. Here 'D' is any similarity measure that is used to compare the closeness between the two vectors. Here, $h_{\mathbf{ts}}$ is the neighborhood function; a non-increasing function of the distance between the 's' node and all the other nodes in the lattice.

- 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) is minimized.

$$\mathbf{e}(\mathbf{v}, \mathbf{w_r}) = \sum_{s} h_{\mathbf{rs}} D(\mathbf{v} - \mathbf{w_s}) \tag{2}$$

  To avoid local minima, $h_{\mathbf{rs}}$ is selected as a convex function, like the middle region of Gaussian function, with a large range at the start and is gradually reduced to a delta function (δ) [9].

- If the similarity measure considered is the Euclidean distance, $D(\mathbf{v} - \mathbf{w_s}) = \|\mathbf{v} - \mathbf{w_s}\|^2$, then the on-line updating rule for the weights is obtained by taking the derivative and minimizing (2). The update is

$$\mathbf{w_s}(n+1) = \mathbf{w_s}(n) + \epsilon h_{\mathbf{rs}}(\mathbf{v}(n) - \mathbf{w_s}(n)) \tag{3}$$

As discussed before, 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 the neighborhood function is best summarized as described in [11]: the reason for a large neighborhood is to correlate the direction of weight updates of a large number of weights around the BMU, $\mathbf{r}$. As the range decreases, so does the number of neurons correlated in the same direction. This correlation ensures that similar inputs are mapped together and hence, the topology is preserved.

2.2. Energy Function and Batch Mode

Heskes [13] has shown the above algorithm for learning SOM minimizes a well defined energy function. With finite number fo samples, the energy function is given by (in the discrete case)

$$E(W) = \sum_{n}^{N} \sum_{s}^{M} h_{\mathbf{rs}} D(\mathbf{v}(n) - \mathbf{w_s}) \tag{4}$$

To find the batch mode update rule, we can take the derivative of $E(W)$ w.r.t $\mathbf{w_s}$ 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

$$\mathbf{w_s}(n+1) = \frac{\sum_n^N h_{\mathbf{rs}} \mathbf{v}(n)}{\sum_n^N h_{\mathbf{rs}}}$$

2.3. Other variants

Contrary to what is assumed in the SOM-MSE case, 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. [24] that in a continuum mapping, i.e., having infinite neighborhood node density, and a 1D map developed in a one dimensional input space (or multidimensional space which are separable) the weight density $P(\mathbf{w}) \propto P(\mathbf{v})^{2/3}$. When a

discrete lattice is used there is a correction in the relation given by

$$\begin{aligned} p(w) &= p(v)^{\alpha} \\ with \quad \alpha &= \frac{2}{3} - \frac{1}{3\sigma_h^2 + 3(\sigma_h + 1)^2} \end{aligned} \tag{5}$$

where σ_h is the neighborhood range in case of a rectangular function. There are several other methods [31] proposed with different definitions of neighborhood function, but the mapping is unable to produce an optimal mapping, i.e, with a magnification of 1. We observed that is all these cases, the weights are always oversampling the low probability regions and undersampling 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 the 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, oversampling that region.

By slightly modifying the updating rule, Bauer and Der [5] and Claussen [7] have proposed different methods to obtain a mapping with magnification factor of 1. Bauer and Der [5] have used the local input density at the weights to adaptively control the step size of the weight update. Such a 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 [31]. Likewise, the method proposed by Claussen [7] is not able to produce a stable result in case of high dimensional data.

A completely different approach is taken by Linsker [18], where mu-

tual information between the input density and the weight density is maximized. It is shown that such 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 only for Gaussian distributions. Furthermore, Van Hulle [31] has proposed another information-theoretic algorithm based algorithm based on Bell and Sejnowski [6]'s Infomax principle, where the differential entropy of the output nodes is maximized.

In the recent past, inspired by the use of *kernel Hilbert spaces* by Vapnik [32], several kernel based topographic mapping algorithms are proposed [2, 10, 20]. Graepel [10] has used the theory of deterministic annealing [25] to develop a new self organizing network called the soft topographic vector quantization (STVQ). A kernel based STVQ (STMK) has also been proposed where the weights are considered in the feature space rather than the input space. To overcome this difficulty, Andras [2] proposed a kernel-Kohonen network in which the input space is transformed, both the inputs and the weights, into a high-dimensional reproducing kernel Hilbert space (RKHS) and the Euclidian distance in the high-dimensional space is the cost function to update the weights. Analyzed in the context of classification, the idea comes from the theory of non-linear support vector machines [11] 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.

Lau et al. [16] 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 original Kohonen network. Moreover, Yin [36] has compared KSOM with the self organizing mixture networks (SOMN) [35] and showed that KSOM is equivalent to SOMN, and in turn to the SOM itself. However, they conclude that the choice of the kernel function and their parameters does significantly effect the performance of the model.

In this work, we study the KSOM from the perspective of information theoretic learning [22, 23] and give further insight into performance and nature of the KSOM. We first analyze KSOM using the properties of the Correntropy Induced Metrix (CIM) [19, 26, 34] and then show how the shape of the kernel function influences the mapping. More precisely, we show that using kernel functions that have strong outlier rejection properties, leads to a mapping with better magnification factor and hence, captures the input probability distribution more accurately. Also, this further gives insight into the relationship between the KSOM and Parzen density estimation. We leverage this property to find the relationship between the energy function of KSOM and information theoretic learning [22, 23] quantities, like KL-divergence or cross entropy, to learn the parameters of the kernel function, particularly the kernel bandwidth of the radial basis function.

2.4. Correntropy and its properties

Correntropy is a generalized similarity measure between two random variables X and Y defined in [19] as:

$$V_\sigma(X,Y) = E[\kappa_\sigma(X-Y)] \tag{6}$$

Here we use the Gaussian kernel

$$G_\sigma(\mathbf{x},\mathbf{y}) = \frac{1}{(\sqrt{2\pi}\sigma)^d} exp\bigg(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\bigg) \tag{7}$$

where d is the input dimensions and σ is the *kernel bandwidth*, since it is most popularly used in the information theoretic learning (ITL) literature. Another popular kernel is the Cauchy kernel which is given by

$$C_\sigma(\mathbf{x}, \mathbf{y}) = \frac{\sigma}{\sigma^2 + \|\mathbf{x} - \mathbf{y}\|^2} \tag{8}$$

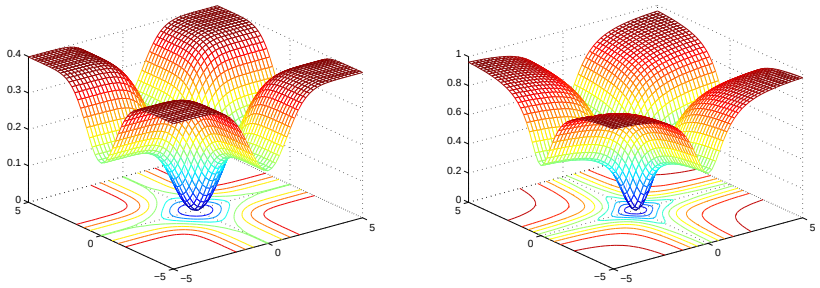
In practice, only a finite number of samples of the data are available and hence, correntropy can be estimated as

$$\hat{V}_{N,\sigma} = \frac{1}{N} \sum_{i=1}^{N} \kappa_\sigma(\mathbf{x}_i - \mathbf{y}_i)$$

One of the important properties of correntropy is that it induces a metric called the correntropy induced metric (CIM) in the sample space [19, 26]. Given two sample vectors $\{x_1, x_2, ..., x_N\}$ and $\{y_1, y_2, ..., y_N\}$, CIM is defined as

$$\begin{aligned} CIM(X, Y) &= (\kappa_\sigma(0) - \hat{V}_\sigma(X, Y))^{1/2} \\ &= \left(\frac{1}{N} \sum_{n}^{N} \kappa_\sigma(0) - \kappa_\sigma(x_n - y_n) \right)^{1/2} \end{aligned} \tag{9}$$

It has been observed that the CIM induces a non-linear metric, whose shape is dependent on the kernel function. In case of Gaussian and Cauchy kernel, the metric behaves like an L2 norm when the two vectors are close. The CIM behaves like the L1 norm for more distant vectors, and eventually becomes insensitive to the distance between the two vectors, as the L0 norm does. The extent of the space over which the CIM acts as the L2 or L0 norm is directly related to the kernel bandwidth, σ. 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 metric.



(a) with Gaussian Kernel (b) with Cauchy Kernel

Figure 1: Surface plot CIM(X,0) in 2D sample space. (kernel width is 1)

Figure 1 demonstrates the non-linear nature of the surface of the CIM, with N=2. Note that the shape of the L2 norm depends on the kernel function and the kernel bandwidth in turn determines the extent of the L2 regions. As we will discuss later, these parameters play an important role in determining the quality of the final output.

3. SOM with correntropy induced metric

As discussed above, correntropy induces a non-linear metric in the input space called the 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.

If $\mathbf{v}(n)$ is considered to be the input vector at a time instant n, then the best matching unit (BMU) can be obtained using the CIM as

$$\begin{aligned} \mathbf{r} &= \arg\min_{s} \sum_{t} h_{\mathbf{ts}} CIM(\mathbf{v}(n) - \mathbf{w_t})_{N=1} \\ &= \arg\min_{\mathbf{s}} \sum_{t} h_{\mathbf{ts}} (\kappa_\sigma(0) - \kappa_\sigma(\|\mathbf{v}(n) - \mathbf{w_t}\|)) \end{aligned}$$

where $\mathbf{r}$ is the BMU at instant n. This stochastic approximation of CIM can be seen as being similar to kernel trick used to train the KSOM in [2]. Henceforth, we call this model SOM-CIM instead of KSOM.

Following the same procedure as described in section 2.1, the update rule can be written as:

$$\mathbf{w_s}(n+1) \quad = \quad \mathbf{w_s}(n) - \eta \triangle \mathbf{w_s} \tag{10}$$

If the Gaussian function is used, then the gradient is

$$\triangle \mathbf{w_s} \quad = \quad -h_{\mathbf{rs}} G_\sigma(\|\mathbf{v}(n) - \mathbf{w_s}\|)(\mathbf{v}(n) - \mathbf{w_s}) \tag{11}$$

In Cauchy kernel case, it is

$$\triangle \mathbf{w_s} \quad = \quad -h_{\mathbf{rs}} \frac{1}{(\sigma^2 + \|\mathbf{v}(n) - \mathbf{w_s}\|^2)}(\mathbf{v}(n) - \mathbf{w_s}) \tag{12}$$

The σ terms in (11) and (12), which are left implicit, are combined with the learning rate η, which is a free parameter.

In the batch mode, when a finite number of input samples are present the overall cost function becomes

$$E_{CIM}(W) = \sum_n^N \sum_s^M h_{\mathbf{rs}} CIM(\mathbf{v}(n) - \mathbf{w_s}) \tag{13}$$

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

$$\begin{aligned} \frac{\partial E_{CIM}(W)}{\partial \mathbf{w_s}} \quad &= \quad -\sum_n^N h_{\mathbf{rs}} G_\sigma(\mathbf{v}(n) - \mathbf{w_s})(\mathbf{v}(n) - \mathbf{w_s}) = 0 \\ \Rightarrow \mathbf{w_s} \quad &= \quad \frac{\sum_n^N h_{\mathbf{rs}} G_\sigma(\mathbf{v}(n) - \mathbf{w_s})\mathbf{v}(n)}{\sum_n^N h_{\mathbf{rs}} G_\sigma(\mathbf{v}(n) - \mathbf{w_s})} \end{aligned}$$

This update is the iterative fixed point update rule, indicating that the weights are updated iteratively and can not reach minima in one iteration, as it does with the MSE.

In the Cauchy kernel case, the update rule is

$$\mathbf{w_s} = \frac{\sum_n^N h_{\mathbf{rs}} C_\sigma(\mathbf{v}(n) - \mathbf{w_s})\mathbf{v}(n)}{\sum_n^N h_{\mathbf{rs}} C_\sigma(\mathbf{v}(n) - \mathbf{w_s})}$$

3.1. Magnification Factor and Relationship with Density Estimation

We observe that in case of both Gaussian and Cauchy kernels, compared to the update rule while using the MSE (3), the gradient to update the weights of SOM-CIM has an additional scaling factor. This additional scaling factor, whose value is small when $\|\mathbf{v}(n) - \mathbf{w_s}\|$ is large, in the updating rule points out the strong outlier rejection (or less influenced by the low probability samples) capability of the CIM. So, with the appropriate choice of the kernel bandwidth σ, this property of the CIM is able to overcome one of the key problems associated with using MSE, i.e., oversampling 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) emphasis more on the higher probability regions and hence, 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, σ. On the other hand, when the kernel functions that does not have this property, like the polynomial kernel, the KSOM might not produce a better solution. This explain how the choice of the kernel function influences the performance of KSOM.

Another property of the CIM (with a Gaussian kernel) that influences the magnification factor is the presence of higher order moments. According

to Zador [37], the order of the error directly influences the magnification factor of the mapping. Again, since the kernel bandwidth, σ, determines the influence of the higher order moments (refer to section 2.4) the choice of σ plays an important role in the formation of the final mapping.

Also, it is interesting if we look back at the cost function:

$$\begin{aligned} E_{CIM}(W) &= \sum_{n}^{N}\sum_{\mathbf{s}}^{M} h_{\mathbf{rs}}(1-\kappa_\sigma(\mathbf{w_s},\mathbf{v}(n))) \\ &= \sum_{n}^{N}\sum_{\mathbf{s}}^{M} h_{\mathbf{rs}} - \sum_{\mathbf{s}}^{M} h_{\mathbf{rs}}\kappa_\sigma(\mathbf{w_s},\mathbf{v}(n)) \end{aligned} \tag{14}$$

where the second term in (14) can be considered to be the estimate of the probability of the inputs when the kernel function is of density type. In case of a radial basis function, this is equivalent to using the sum of the Gaussian mixtures centered at the weights, with the neighborhood function considered equivalent to the posterior probability [36]. More formally, this mixture model can be written as:

$$p(\mathbf{v}(n)|\mathbf{W}) \approx \sum_{i} p(\mathbf{v}(n)|\mathbf{w_i})P(\mathbf{w_i})$$

Hence, it can be said that this way of formulating the SOM is equivalent to estimating the input probability distribution.

However, it can also be considered to be the weighted Parzen density estimation [4] technique, with the neighborhood function acting as the strength associated with each weight vector. In fact, energy function in (14) is equivalent to the cross information potential (CIP) (derived from the Renyi's definition of cross-entropy [22]) and quantifies the similarity between two probability distribution. Hence, by minimizing the energy function in (14), we can expect an information-theoretically optimal mapping. This reasserts

the argument that the SOM-CIM can increase the magnification factor of the mapping to 1 and also gives insight into the choice of the kernel function.

Another important point regarding the kernel bandwidth should also be considered here. If the value of σ is small, then during the 'ordering phase' [11] 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 in our simulations, a large σ 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.

3.2. Results

3.2.1. Magnification factor of the SOM-CIM

As pointed out earlier, the use of the CIM does not allow the nodes to oversample the low probability regions of the input distribution as they do with the MSE. Also, the presence of the higher order moments affect the final mapping. So, 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. [24], to demonstrate the magnification of the SOM using the MSE.

Here, a one-dimensional input space is mapped onto a one dimensional map. Specifically, $100,000$ input instances drawn from the distribution $f(x) = 2x$ are mapped onto a 50 node one dimensional *chain*. A Gaussian neighborhood function is considered and its range is decreased with the

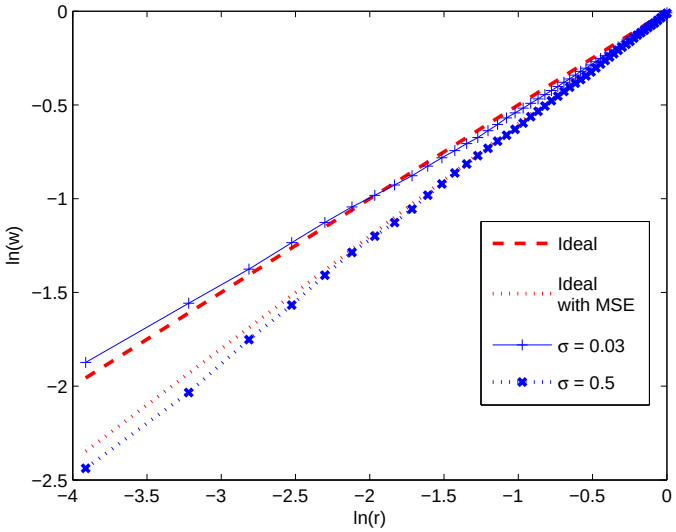


Figure 2: The figure shows the plot between $ln(\mathbf{r})$ and $ln(\mathbf{w})$ for different values of σ. '- -' indicates the ideal plot representing the relation between $\mathbf{w}$ and $\mathbf{r}$ when the magnification is 1 and '...' indicates the ideal plot representing the magnification of 2/3. '-+-' indicates the relation obtained with SOM-CIM when $\sigma = 0.03$ and '-x-' indicates when $\sigma = 0.5$.

number of epochs as follows:

$$\begin{aligned} h_{rs}(n) &= exp\Big(-\frac{(r-s)^2}{2\sigma_h(n)}\Big) \\ \text{where} \quad \sigma_h(n) &= \sigma_h^i exp\Big(-\theta * \sigma_h^i \frac{n}{N}\Big) \end{aligned} \tag{15}$$

where $\mathbf{r}$ and $\mathbf{s}$ are node indices, σ_h is the range of the neighborhood function, σ_h^i = initial value of σ_h, $\theta = 0.3$ and N = number of iterations/epochs.

If the magnification factor is '1', indicating an optimal mapping, the relation between the weights and the nodes is $ln(w) = \frac{1}{2}ln(r)$. In the case of the SOM with the MSE (SOM-MSE) as the cost function, the magnification is proved to be 2/3 and the relation comes out to be $ln(w) = \frac{3}{5}ln(r)$. Fig 2 shows that when a smaller σ is considered for the SOM-CIM, then a better

magnification can be achieved. As the value of the σ 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 2.4, that with the increase in the value of σ the surface tends to behave more like MSE. Actually, it can be said that by varying the value of σ the magnification factor of the SOM can vary between 2/3 and 1!

3.2.2. Maximum Entropy Mapping

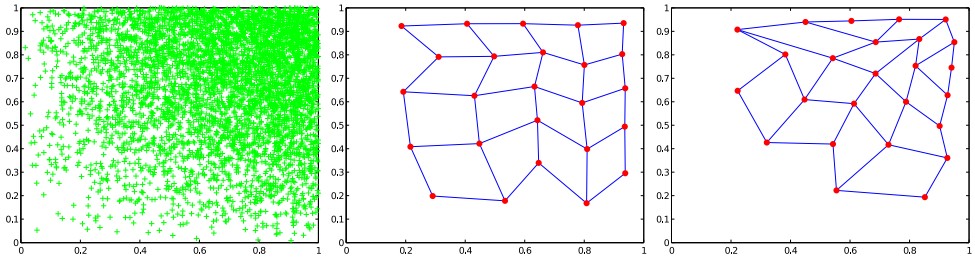
Optimal information transfer from the input distribution to the weights happens when the magnification factor is 1. Such a mapping should ensure that every node is active with equal probability, also called the equiprobabilistic mapping [31]. Since the entropy is a measure that determines the amount of information content, we use Shannon's entropy (16) of the weights to quantify this property.

$$I = -\sum_{\mathbf{r}=1}^{M} p(\mathbf{r}) ln(p(\mathbf{r})) \tag{16}$$

where $p(\mathbf{r})$ is the probability of the node r to be the winner.

Table 1 shows the change in I 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. We choose the number of epochs to be 50 and with neighborhood function parameter going from $\sigma_h = 10 \rightarrow 0.0045$.

From table 1, we observe that the value of the σ influences the quality of the mapping. Figure 3 shows the weight distribution at the convergence after 50 epoches. We also observe, for large σ, because of the wider L2



(a) Input scatter plot (b) SOM - MSE (c) SOM - CIM

Figure 3: The figures show 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.

region, the mapping of the SOM-CIM behaves similarly to that of the SOM-MSE. But as σ is decreased, the nodes try to be equiprobabilistic indicating that they are not oversampling the low probability regions. Setting the value of σ too small distorts the mapping. This is due to the restricted movement of the nodes, which leads to oversampling the high probability region. This underlines the importance of the value of σ to obtain a good quality mapping. 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 σ, indicating that the shapes 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

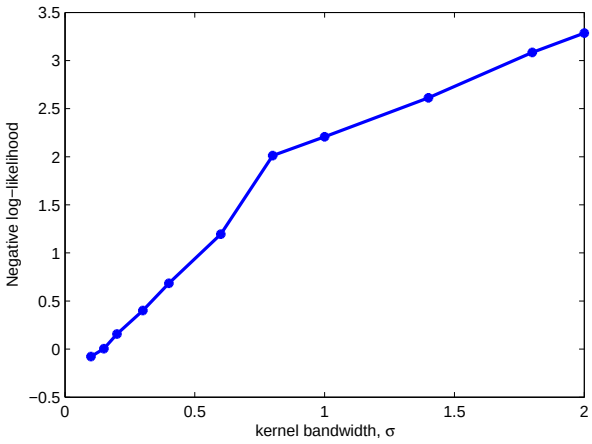


Figure 4: Negative log likelihood of the input versus the kernel bandwidth. The figure shows how the negative log likelihood of the inputs given the weights change with the change in the kernel bandwidth. Note that for negative value of the negative log-likelihood in the plot indicates that the corresponding values of the model, the weights and the kernel bandwidth, are not appropriate to estimate the input distribution.

given by

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

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

The change in its value with the change in the kernel bandwidth is plotted in the figure 4. For an appropriate value of the kernel bandwidth, the likelihood of estimating the input is high; whereas for large values of σ it decreases considerably.

4. SOM-CIM with Adaptive Kernels

Now that the influence of the kernel bandwidth on the mapping is studied, one should understand the difficulty involved in setting the value of σ.

Though it is understood from the theory of Correntropy [19] that the value of σ 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 [27] can be used. But it is observed that these results are suboptimal in many cases because of the Gaussian distribution assumptions put on the underlying data. It should also be noted that for equiprobable 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.

4.1. The Algorithm

Clustering of the data is closely related to 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 [17], Vector Quantization using KL- Divergence [12] and Principle of Relevant Information [22], 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 in 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. [8] of using the Kullback-Leibler diver-

gence (D_{KL}) to estimate the kernel bandwidth for density estimation and extend it to the clustering algorithms.

The $D_{KL}(f\|g)$ can be estimated non-parametrically from the data using Parzen window estimation. Here, if f is the estimated pdf using the data points $\mathbf{v}_i, i \in \{1, 2, ..., N\}$ and g is the estimated pdf using the weight vectors $\mathbf{w}_j, j \in \{1, 2, ..., M\}$, then the D_{KL} can be written as

$$D_{KL}(f, g) = E_f\left[log(\sum_i^N G_{\sigma_v}(\mathbf{v} - \mathbf{v}_i))\right] - E_f\left[log(\sum_j^M G_{\sigma_w}(\mathbf{v} - \mathbf{w}_j))\right] \quad (17)$$

where σ_v is the kernel bandwidth for estimating the pdf using the input data and σ_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 (17). Hence, the cost function for adapting the kernel bandwidth is

$$J(\sigma) = -\sum_n^N \left[log(\sum_j^M G_{\sigma_w}(\mathbf{v}(n) - \mathbf{w}(j)))\right] \quad (18)$$

where expectation of f is replaced by summation. This is also called *Shannon's cross entropy.*

Now, the estimation of the pdf using the weight vector can be done using a single kernel bandwidth for all the weight vectors, called the homoscedastic case, or using different kernel bandwidths for each weight vector, called the heteroscedastic case. In each of these cases, the kernel bandwidth(s) are obtained using gradient descent over $J(\sigma)$.

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

- The winner is selected using the local error as

$$\mathbf{r} = \arg\min_{\mathbf{s}} \sum_{\mathbf{t}}^M h_{\mathbf{st}}(G_{\sigma(n)}(0) - G_{\sigma(n)}(\|\mathbf{v}(n) - \mathbf{w_t}\|))$$

- The weights and then the kernel bandwidth are updated as

$$\begin{aligned} \triangle \mathbf{w_s} &= -h_{\mathbf{rs}} G_{\sigma(n)}(\|\mathbf{v}(n) - \mathbf{w_s}\|) \frac{(\mathbf{v}(n) - \mathbf{w_s})}{\sigma(n)^3} \\ \mathbf{w_s}(n+1) &= \mathbf{w_s}(n) - \eta \triangle \mathbf{w_s} \end{aligned}$$

$$\begin{aligned} \triangle \sigma(n) &= -\left\{ \frac{\sum_j^M G_{\sigma_(n)}(\mathbf{v}(n) - \mathbf{w}_j(n+1) \left[\frac{\|\mathbf{v}(n) - \mathbf{w}_j(n+1)\|^2}{\sigma_(n)^3} - \frac{d}{\sigma_(n)} \right]}{\sum_j^M G_{\sigma_(n)}(\mathbf{v}(n) - \mathbf{w}_j(n+1)} \right\} \\ \sigma(n+1) &= \sigma(n) - \eta_\sigma \triangle \sigma(n) \end{aligned}$$

- In batch mode, the weights and the kernel bandwidth are updated as

$$\begin{aligned} \mathbf{w_s}^+ &= \frac{\sum_n^N h_{\mathbf{rs}} G_\sigma(\mathbf{v}(n) - \mathbf{w_s}) \mathbf{v}(n)}{\sum_n^N h_{\mathbf{rs}} G_\sigma(\mathbf{v}(n) - \mathbf{w_s})} \\ \sigma^+ &= \frac{1}{Nd} \sum_n^N \frac{\sum_j^M G_\sigma(\mathbf{v}(n) - \mathbf{w}_j) \left[\|\mathbf{v}(n) - \mathbf{w}_j\|^2 \right]}{\sum_j^M G_\sigma(\mathbf{v}(n) - \mathbf{w}_j)} \end{aligned}$$

In case of heteroscedastic kernels, the same update rules apply for the weights but with σ specified for each node. In this case, the kernel bandwidths are updated as

On-line mode:

$$\begin{aligned} \triangle \sigma_i(n) &= -\left\{ \frac{G_{\sigma i(n)}(\mathbf{v}(n) - \mathbf{w}_i(n+1)) \left[\frac{\|\mathbf{v}(n) - \mathbf{w}_i(n+1)\|^2}{\sigma_i(n)^3} - \frac{d}{\sigma_i(n)} \right]}{\sum_j^M G_{\sigma j(n)}(\mathbf{v}(n) - \mathbf{w}_j(n+1)} \right\} \\ \sigma_i(n+1) &= \sigma_i(n) - \eta_\sigma \triangle \sigma_i(n) \end{aligned}$$

Batch mode:

$$\sigma_i^+ = \frac{1}{d} \frac{\sum_n^N \frac{G_{\sigma i}(\mathbf{v}(n) - \mathbf{w}_i) \left[\|\mathbf{v}(n) - \mathbf{w}_i\|^2 \right]}{\sum_j^M G_\sigma(\mathbf{v}(n) - \mathbf{w}_j)}}{\sum_n^N \frac{G_{\sigma_w(i)}(\mathbf{v}(n) - \mathbf{w}(i))}{\sum_j^M G_{\sigma_w(j)}(\mathbf{v}(n) - \mathbf{w}(j))}}$$

However, as observed in [29] that when the input has high density parts, the kernel bandwidth of the weights representing these shrinks too small

to give any sensible similarity measure and makes the system unstable. To counter this, a scaling factor ρ can be introduced as

$$\triangle\sigma_i(n) \quad = \quad -\left\{ \frac{G_{\sigma i(n)}(\mathbf{v}(n) - \mathbf{w}_i(n+1)) \left[\frac{\|\mathbf{v}(n) - \mathbf{w}_i(n+1)\|^2}{\sigma_i(n)^3} - \frac{\rho d}{\sigma_i(n)} \right]}{\sum_j^M G_{\sigma j(n)}(\mathbf{v}(n) - \mathbf{w}_j(n+1)} \right\}$$

This ensures that the value of σ does not become 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.

4.2. Relationship with Other Methods

Yin and Allinson [35] and Van Hulle [28] also proposed topographic mapping methods with kernel bandwidth estimation based on KL-divergence, called as self-organizing mixture network (SOMN) and local density modeling (LDE), respectively. However, both these methods do so using a Gaussian mixture model and approximate the cost function using Bayesian statistics; heuristically assuming that the neighborhood function acts as mixing parameters of the Gaussian mixture model [29]. By using this single cost function for updating both the centers and the kernel bandwidth, we observe that, as the neighborhood function shrinks at convergence, the kernel bandwidth become too small and does not lead to a good solution. In order to avoid this, the neighborhood functions needs to be slowly annealed leading to slower convergence rate overall. In addition, Yin and Allinson's Gaussian mixture model (SOMN) has limited lattice unfolding capability and does not lead to a good topographic mapping [30]. Hence, is not discussed further.

On the other hand, we adopt a different approach here, where we consider two different cost functions: one for updating the centers $\mathbf{w}$, which is influenced by the neighborhood function, and another for updating the kernel bandwidth, which is independent of the neighborhood function. As

we will show, this allows the proposed approach to model the underlying distribution more accurately, as the kernel bandwidth does not shrink to smaller values and model the local distribution more accurately.

4.3. Results

We first show the performance on an experiment similar to that in section 3.2.1, where $100,000$ samples from the distribution $f(x) = 2x$ are mapped onto a 1D chain of 50 nodes in on-line mode and the relation between the weights and the node indices is observed. Figure 5 shows the results obtained. We observe that in the homoscedastic case the mapping converges to the ideal mapping. On the other hand, in case of the heteroscedastic kernels, the system becomes unstable for $\rho = 1$. To obtain a stable solution, ρ is set to 0.7. This stabilizes the mapping by not allowing the kernel bandwidth to shrink too small. However, it increases the kernel bandwidth of the nodes in low probability region, and hence, distorting the mapping as shown in 5(d).

It is also interesting to see the changes in the kernel bandwidth during learning as shown in the figure 5(a) and figure 5(b). Initially, all the kernels converge to the same bandwidth as the nodes are concentrated at the center. As the 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 3, can be resolved by having relatively larger bandwidths at the beginning, allowing the free movement of the nodes during the ordering phase.

[1]The best result from Table 1 is reproduced for comparison.

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 weights. Results obtained for the experiment similar to the one in section 3.2.2, 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 2. 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 1 ($\sigma = 0.1$) and at the same time is able to transfer more information to the weights. In the heteroscedastic case the system is unstable with $\rho = 1$ and fails to unfold, concentrating more on the high probability regions. On the other hand, when ρ is set to 0.5, the resulting map does not provide a good output because of the large σ values. Figure 6 clearly demonstrates this. When $\rho = 0.5$, it clearly oversample's the low probability regions because of the large kernel bandwidths of the nodes representing them.

For comparison, we also show the performance of LDE [28] on the same task. We note that, in our simulations LDE is slow to converge and requires 200 epochs with slow annealing of the neighborhood function to obtain a stable solution. Moreover, LDE also contains a parameter ρ to find a stable kernel width. However, the recommended value $\rho = 0.4$ [28] led to instability during convergence and we set the value of $\rho = 0.02$ after performing a parameter sweep to obtain the best performance. As shown in Table 2, the proposed model performance better than LDE, in terms of both MSE and information content (or entropy).

5. Experiments

Now that is has been shown how to adapt the SOM using the CIM, we apply the SOM-CIM in a few common applications of the SOM, like density estimation, clustering and principal curves, and show how it can improve the performance.

5.1. 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 [3] and clustering is used to find the density estimation [31]. 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 Parzen non-parametric density estimation procedure [21].

In this experiment, we try to estimate a 2-dimensional Laplacian density function shown in Figure 7(a). The choice of the Laplacian distribution is appropriate to compare these methods because it is a heavy tail distribution and hence, is difficult to estimate without the proper choice of the kernel bandwidth.

Figure 7 and Table 3 shows the results obtained when different methods are used to estimate the input density from the learned weights. When the SOM-MSE is used, the kernel bandwidth is estimated using Silverman's rule [27]:

$$\sigma = 1.06\sigma_f N^{-5}$$

where σ_f is the variance of the input and N is the number of input samples. We observe, as demonstrated in Figure 7(b), that this procedure is not able to produce a good result because of the large number of bumps in the low

probability regions. This is due to the oversampling of the low probability regions when MSE is used.

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. 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, is able to reduce the divergence between the true and estimated densities.

5.2. Principal Surfaces and Clustering

5.2.1. Principal Surfaces

Topology preservation maps can be interpreted as an approximation procedure for the computation of principal curves, surfaces or higher-dimensional principal manifolds [24]. 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 is the definition of principal surfaces in the 1-dimensional case, and can be generalized to high dimensional principal manifolds as [24]:

> Let $f(s)$ be a surface in the vector space V, i.e, $dim(f) - dim(V) - 1$, and let $d_f(\mathbf{v})$ be the shortest distance of a point $\mathbf{v} \in V$ to the

surface f. f is a principal surface corresponding to the density distribution $P(\mathbf{v})$ in V, if the "*mean squared distance*"

$$D_f = \int d_f^2(\mathbf{v}) P(\mathbf{v}) d^L\mathbf{v}$$

is extremal with respect to local variation of the surface.

But the use of the MSE as the criteria for the goodness of the principal surfaces, makes it weakly defined since only the second order moments are used and also because of the distortion in the mapping when outliers are present. Figure 8(a) 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 is mitigated and gives a better approximation of the principal surfaces. Figure 8 illustrates this in case of the SOM-CIM with homoscedastic kernels, where the kernel bandwidth adapts such that the outliers do not have significant effect on the final mapping throughout learning.

5.2.2. Avoiding Dead Units

Another problem with the SOM is that it can yield nodes that are never active, called *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 [31]. 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 9(a) 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 3.2.2, the mapping tries to be equiprobabilistic and hence, avoids dead units (Figure 9(b)).

Table 4 show the results obtained over three different datasets mapped onto a 10×5 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 [1].

The Iris dataset contains 3 classes of 50 instances each, where each class refers to a type of iris plant. 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 is therefore able to give a better visualization of the data.

Another important observation is that the adaptive kernel algorithms are unable to give a good result for both Iris and Blood Transfusion datasets. The reason for this might be the nature of Parzen density estimation, which is the center piece for this algorithm. As the number of dimensions increases, the number of input samples required by 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. Also, allowing multivariate kernels Erdogmus et al. [8] (instead of univariate kernels as in the proposed model) might provide greater flexibility to model complex distributions like in this case.

6. Conclusion

The use of the kernel Hilbert spaces by Vapnik [32] 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. [16], Andras [2], MacDonald [20], etc. In all these cases, the use of the kernel trick is viewed in context of pattern classification and the increase in the 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 final output of the mapping is affected by the choice of the kernel and kernel bandwidth in the cost function. As indicated, the choice of the kernel bandwidth dictates the magnification of the mapping. For example, larger bandwidths cause the SOM-CIM to minimize the quantization error because of the large L2-norm induced, where as a smaller bandwidth might produce a map that concentrates more on the high probability parts, and thus, distorting the mapping. As previously discussed, the advantage of using the CIM lies in its strong outlier rejection capability and the presence of higher order moments. Both these properties are useful when the input distribution is non-uniform and 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. It is observed that the algorithm is unstable in the heteroscedastic case and an additional free parameter ρ needs to be specified. In spite of that, the final mapping is still less sensitive to the value of ρ than the value of σ 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. 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, although the dependence of the magnification factor on the kernel bandwidth is shown experimentally, the theoretical analysis is still elusive. The future work should concentrate on finding the relation between the magnification factor and the kernel bandwidth, which in turn depends on the variance of the data. 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 multivariate kernels using the idea proposed by Erdogmus et al. [8] but will lead to a larger computational complexity. An effective, less computationally expensive algorithm to adapt the kernel bandwidth is necessary.

Acknowledgement

This work is supported by the Office of Naval Research (ONR) grant #N000141010375. We thank Evan Kriminger for his valuable comments and suggestions.

References

[1] D.J. Newman A. Asuncion. UCI machine learning repository, 2007. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[2] P. Andras. Kernel-kohonen networks. *International Journal of Neural Systems*, 12:117–135, April 2002.

[3] Adelchi Azzalini and Nicola Torelli. Clustering via nonparametric density estimation. *Statistics and Computing*, 17:71–80, 2007.

[4] Gregory A. Babich and Octavia I. Camps. Weighted parzen windows for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:567–570, 1996.

[5] H. U. Bauer and R. Der. Controlling the magnification factor of self-organizing feature maps. *Neural Comput.*, 8(4):757–771, 1996. ISSN 0899-7667.

[6] Anthony J. Bell and Terrence J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Comput.*, 7(6):1129–1159, 1995. ISSN 0899-7667.

[7] Jens Christian Claussen. Winner-relaxing self-organizing maps. *Neural Comput.*, 17(5):996–1009, 2005. ISSN 0899-7667. doi: http://dx.doi.org/10.1162/0899766053491922.

[8] Deniz Erdogmus, Robert Jenssen, Yadunandana N. Rao, and Jose C.Principe. Gaussianization: An efficient multivariate density estimation technique for statistical signal processing. *The Journal of VLSI Signal Processing*, 45:67 –83, 2006.

[9] E. Erwin, K. Obermayer, and K. Schulten. Self-organizing maps: Ordering, convergence properties and energy functions. *Biological Cybernetics*, 67:47–55, 1992.

[10] T. Graepel. Self-organizing maps: Generalizations and new optimization techniques. *Neurocomputing*, 21(1-3):173–190, November 1998.

[11] Simon Haykin. *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice Hall, 2 edition, July 1998. ISBN 0132733501.

[12] A. Hegde, D. Erdogmus, T. Lehn-Schiøler, Y. Rao, and J. Principe. Vector-quantization by density matching in the minimum kullback-leibler divergence sense. In *IEEE International Conference on Neural Networks - Conference Proceedings*, volume 1, pages 105–109, 2004.

[13] Tom Heskes. *Energy function for self-organizing maps*. Elsevierl, Amsterdam, 1999.

[14] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59 – 69, 1982.

[15] Teuvo Kohonen. *Self-Organizing Maps (2nd ed)(Springer Series in Information Sciences, 30)*. Springer, 2nd edition, 1997. ISBN 3540620176.

[16] K. Lau, H. Yin, and S. Hubbard. Kernel self-organising maps for classification. *Neurocomputing*, 69(16-18):2033–2040, October 2006.

[17] Tue Lehn-Schioler, Anant Hegde, Deniz Erdogmus, and Jose Principe. Vector quantization using information theoretic concepts. *Natural Computing*, 4(1):39–51, January 2005. ISSN 1567-7818.

[18] Ralph Linsker. How to generate ordered maps by maximizing the mutual information between input and output signals. *Neural Comput.*, 1(3):402–411, 1989. ISSN 0899-7667. doi: http://dx.doi.org/10.1162/neco.1989.1.3.402.

[19] Weifeng Liu, Puskal P. Pokharel, and Jose C. Principe. Correntropy: Properties and applications in non-gaussian signal processing. *IEEE Transactions on Signal Processing*, 55(11):5286–5298, 2007.

[20] Fyfe C. MacDonald, D. The kernel self-organising maps. In *4th int. conf. on knowledge-based intelligence engineering systems and applied technologies*, pages 317–320, 2000.

[21] Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[22] Jose Principe. *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives, (Springer Series in Information Sciences and Statistics)*. Springer, 1nd edition, 2010. ISBN 978-1-4419-1569-6.

[23] Jose C. Principe, Dongxin Xu, Qun Zhao, and John W. Fisher, III. Learning from examples with information theoretic criteria. *J. VLSI Signal Process. Syst.*, 26(1/2):61–77, 2000. ISSN 0922-5773.

[24] Helge Ritter, Thomas Martinetz, and Klaus Schulten. *Neural Computation and Self-Organizing Maps; An Introduction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992. ISBN 0201554437.

[25] Kenneth Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. In *Proceedings of the IEEE*, pages 2210–2239, 1998.

[26] I. Santamaria, P.P. Pokharel, and J.C. Principe. Generalized correlation function: definition, properties, and application to blind equalization. *Signal Processing, IEEE Transactions on*, 54(6):2187–2197, June 2006.

[27] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, April 1986. ISBN 0412246201.

[28] Marc M Van Hulle. Kernel-based topographic map formation by local density modeling. *Neural Computation*, 14(7):1561–1573, 2002.

[29] Marc M Van Hulle. Joint entropy maximization in kernel-based topographic maps. *Neural Computation*, 14(8):1887–1906, 2002.

[30] Marc M Van Hulle. Kernel-based topographic maps: Theory and applications. *Wiley Encyclopedia of Computer Science and Engineering*, 2008.

[31] M.M Van Hulle. *Faithful Representations and topographic maps: From distortion- to information-based self-organization*. New York: Wiley, 2000.

[32] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0-387-94559-8.

[33] Thomas Villmann and Jens Christian Claussen. Magnification control in self-organizing maps and neural gas. *Neural Comput.*, 18(2):446–469, 2006. ISSN 0899-7667.

[34] Jian-Wu Xu, P.P. Pokharel, A.R.C. Paiva, and J.C. Principe. Nonlinear component analysis based on correntropy. In *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, pages 1851–1855, 0-0 2006. doi: 10.1109/IJCNN.2006.246905.

[35] H. Yin and N.M. Allinson. Self-organizing mixture networks for probability density estimation. *Neural Networks, IEEE Transactions on*, 12 (2):405–411, Mar 2001.

[36] Hujun Yin. On the equivalence between kernel self-organising maps and self-organising mixture density networks. *Neural Netw.*, 19(6):780–784, 2006. ISSN 0893-6080. doi: http://dx.doi.org/10.1016/j.neunet.2006.05.007.

[37] P. Zador. Asymptotic quantization error of continuous signals and the quantization dimension. *Information Theory, IEEE Transactions on*, 28(2):139–149, Mar 1982. ISSN 0018-9448.

Table 1: The information content and the mean square quantization error for various values of σ in SOM-CIM and SOM with MSE is shown. $I_{max} = 3.2189$

Method		Mean Square Quantization Error	Entropy or Info Cont., I
KSOM	kernel Bandwidth, σ	$(. * 10^{-3})$	
	0.05	15.361	3.1301
Gaussian	0.1	5.724	**3.2101**
Kernel	0.2	5.082	3.1929
	0.5	5.061	3.1833
	0.8	5.046	3.1794
	1.0	5.037	3.1750
	1.5	5.045	3.1768
	0.02	55.6635	3.1269
	0.05	5.8907	**3.2100**
Cauchy	0.1	5.3305	3.2065
Kernel	0.2	5.1399	3.1923
	0.5	5.1278	3.1816
	1	5.0359	3.1725
	1.5	5.0678	3.1789
MSE	–	**5.0420**	3.1767

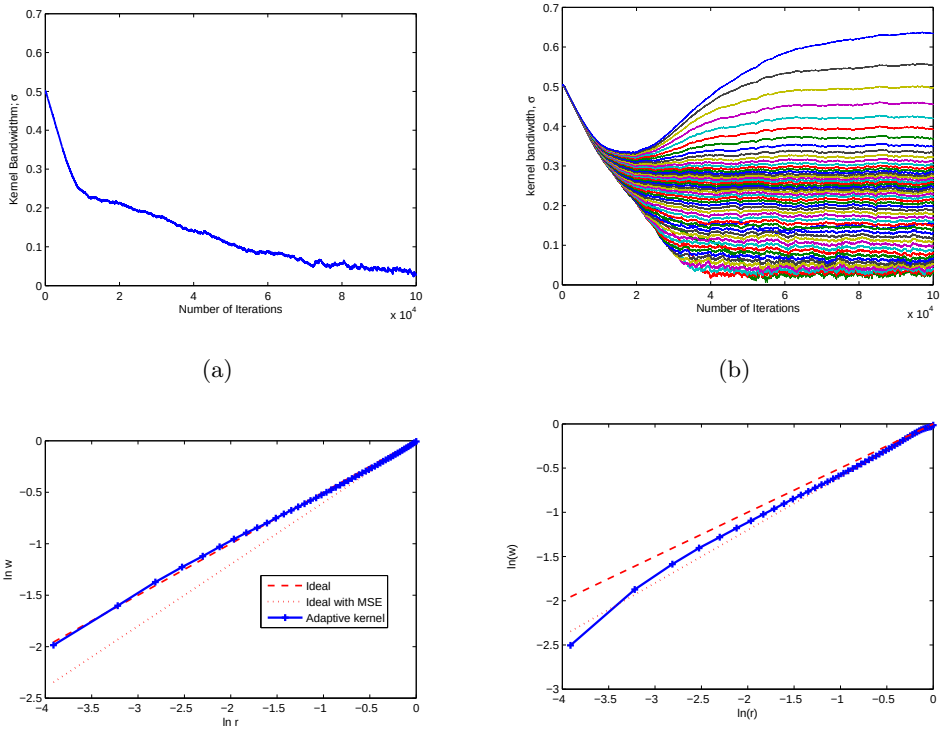
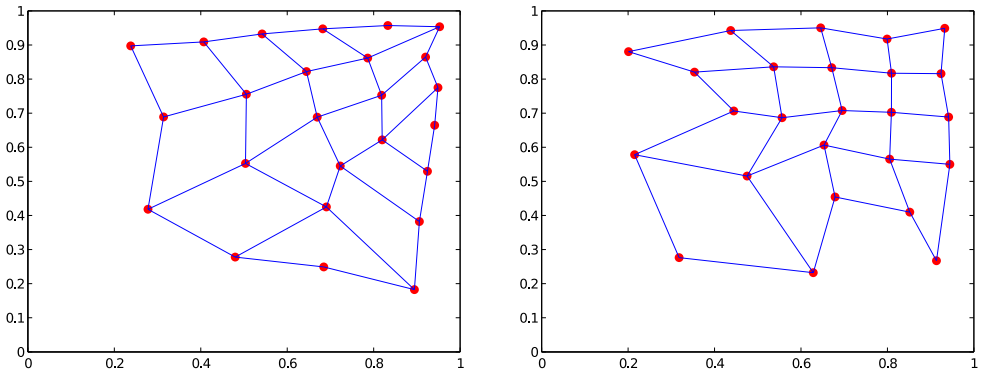


Figure 5: Magnification of the map in the homoscedastic and heteroscedastic cases.The figures [a] and [b] show the tracks of the kernel bandwidth in case of homoscedastic and heteroscedastic kernels, respectively. Figures [c] and [d] show the plot between $ln(\mathbf{r})$ and $ln(\mathbf{w})$. Refer text for explanation.

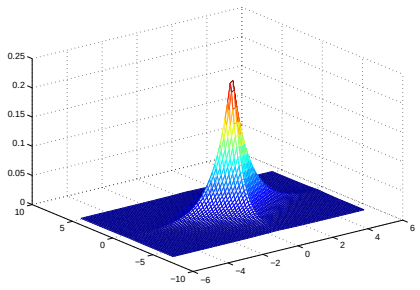
Table 2: The information content and the mean square quantization error for homoscedastic and heteroscedastic cases in SOM-CIM. $I_{max} = 3.2189$

Method	Mean Square Quantization Error	Entropy or Info Cont., I
Homoscedastic kernels	$5.8725 * 10^{-3}$	3.2095
Heteroscedastic kernels ($\rho = 0.5$)	$6.1179 * 10^{-3}$	3.1970
Heteroscedastic kernels ($\rho = 1$)	$15.268 * 10^{-3}$	2.700
Constant kernel [1], $\sigma = 0.1$	$5.7240 * 10^{-3}$	**3.2101**
SOM-MSE	$\mathbf{5.0420 * 10^{-3}}$	3.1767
LDE ($\rho = 0.02$)	$6.250 * 10^{-3}$	3.1813

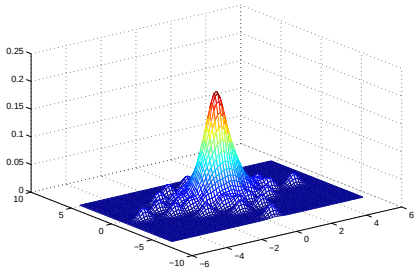


(a) (b)

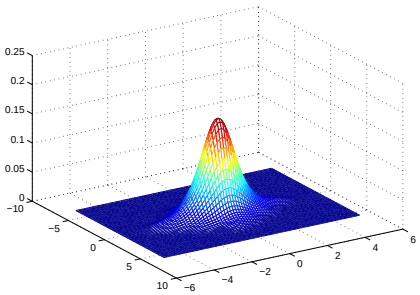
Figure 6: The scatter plot of the weights at convergence in the case of homoscedastic and heteroscedastic kernels. [a] The scatter plot of weights in case of homoscedastic kernels. [b] The scatter plot of weights in the case of heteroscedastic case. The lines indicate the neighborhood in the lattice space. Clearly, the heteroscedastic kernels with $\rho = 0.5$ oversample the low probability regions when compared with the homoscedastic case.



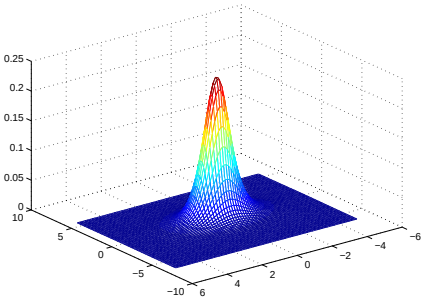
(a) True density function



(b) SOM with MSE



(c) SOM-CIM with homoscedastic kernels

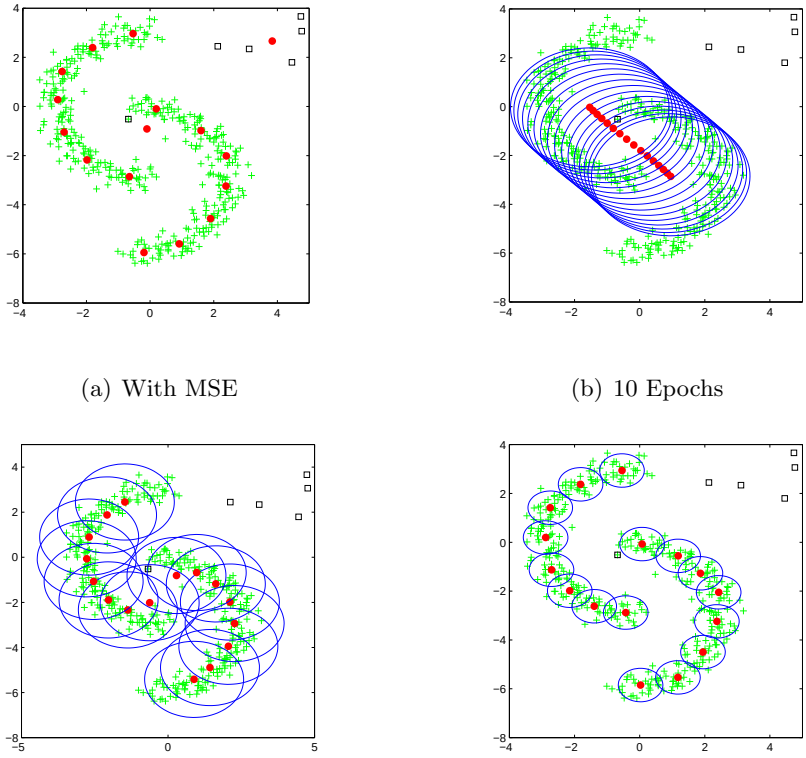


(d) SOM-CIM with heteroscedastic kernels

Figure 7: 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 obtain 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.

Table 3: 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.

Method	MSE $(.*10^{-4})$	KLD
SOM with MSE	2.0854	308.4163
SOM-CIM,		
homoscedastic	2.0189	128.6758
heteroscedastic($\rho = 0.8$)	3.1432	12.6948



(a) With MSE

(b) 10 Epochs

(c) 20 Epochs

(d) 50 Epochs

Figure 8: 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.

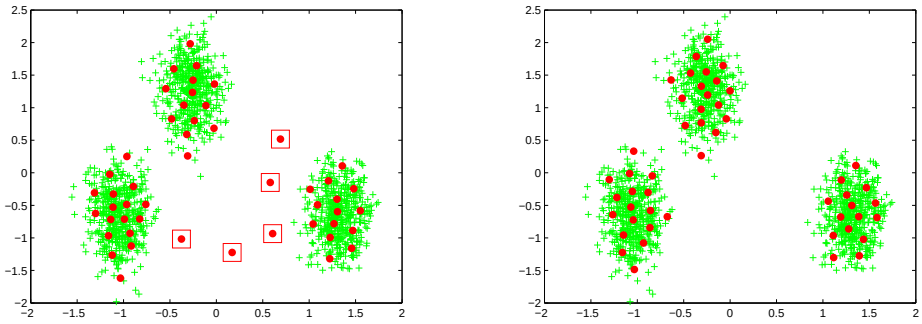


Figure 9: The scatter plot of the weights showing the dead units when CIM and MSE are used to map the SOM. [A] The boxed dots indicate the dead units mapped when MSE is used. [B] Mapping when CIM with $\sigma = 0.5$ is used in the SOM. There are no dead units in this case.

Table 4: Number of dead units yielded for different datasets when MSE and CIM are used for mapping. Each entry is an average over 30 runs.

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