KERNEL SELF-ORGANISING MAPS AND MIXTURE NETWORKS

Hujun Yin and King Wai Lau
School of Electrical and Electronic Engineering, University of Manchester
Manchester, M60 1QD, UK
h.yin@manchester.ac.uk

Abstract – *Kernel methods have been widely applied to various learning models to extend their nonlinear approximation abilities. Such extensions have also recently occurred to the Self-Organising Map (SOM). In this paper, two recent kernel SOMs are reviewed and it is shown that the kernel SOMs can be formally derived from an energy function of the SOM in the feature space. Various kernel functions are readily applicable to the kernel SOM, while their performance and choices of kernel parameters depend on the problem. This paper shows that with an isotropic and density-type kernel function, the kernel SOM is equivalent to a homoscedastic Self-Organising Mixture Network, an entropy-based density estimator. It also explains that the SOM approximates naturally a kernel method.*

Key words – SOM, Kernel Method, Kernel SOM, Mixture Models.

1 Introduction

The Self-Organising Map (SOM) [10] is one of the most popular and widely applied neural network models owing to its several distinct features over other neural networks such as nonlinear mapping of input to output space and topological preserving. The SOM has been studied, applied and extended extensively in the last couple of decades and many insights have been gained since its introduction [10]. However numerous new developments and applications are continuing to emerge [2, 3, 9].

Kernel methods have received a great deal of attention in the past few years, especially in the supervised learning community [16]. By applying kernel function to the input space, a nonlinear, complex problem can become linear in the high dimensional feature space [1]. Typical examples are the Support Vector Machines [5]. Kernel methods have also been applied to unsupervised learning models such as principal component analysis [15], principal factor analysis, projection pursuit and canonical correlation analysis [6]. Two kernel variants of the SOM have been proposed recently. MacDonald and Fyfe [13] derived a kernel SOM from kernelising the *k*-means clustering algorithm with added neighbourhood. Andras [4] and Pan, Chen and Zhang [14] have proposed a kernel SOM by transforming the input space to a feature space using nonlinear kernel functions.

The objectives of these kernel SOMs are different from some earlier approaches [7] and [17, 18], which aim at optimising the topographic mapping and approximating data distribution respectively. In [7], Graepel, Burger and Obermayer apply kernel functions to transform the input to high dimensional space, thus transforming the distance metric to nonlinear and adding more flexibility in vector-quantising and capturing the data structures. In van Hulle [17] and Yin and Allinson [18], neurons in the SOM are treated as Gaussian (or other) kernels, the resulting map approximates a mixture of Gaussian (or other) distribution of the data. It further establishes a link between the mixture model and the self-organisation process [18].

In Section 2 two recent kernel SOMs are reviewed and it is further shown that one of them can be derived from an energy function [11, 8]. Furthermore, we show that the kernel self-organisation can be performed in the transformed space completely. The proposed method unifies the kernel approaches to the SOM. In Section 3, a direct relationship between the kernel SOM and an earlier Self-Organising Mixture Network (SOMN) [18] is revealed. The relation further explains that the kernel SOM is an underlying mixture model and that the SOM is already an approximate of a kernel method. Conclusions are given in Section 4.

2 Kernel Self-Organising Maps

A kernel is a function κ : $X \times X \in$P, where X is the input space. This function is a dot product of mapping function $\phi(\mathbf{x})$, i.e. $\kappa(\mathbf{x};\mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$, where $\phi : X \to F$, F is a high dimensional inner product feature space.

Type I Kernel SOM

Following the kernel PCA [15], a k-means based kernel SOM (type I) has been proposed by MacDonald et. al. [13]. Each data point $\mathbf{x}$ is mapped to the feature space via $\phi(\mathbf{x})$. Each mean can be described as a weighted sum of the mapping functions, $\mathbf{m}_i = \sum_n \gamma_{i,n} \phi(\mathbf{x}_n)$. The algorithm then selects a mean or assigns the data with the minimum distance between the mapped point and the mean,

$$\|\phi(\mathbf{x}) - \mathbf{m}_i\|^2 = \|\phi(\mathbf{x}) - \sum_n \gamma_{i,n} \phi(\mathbf{x}_n)\|^2 = \kappa(\mathbf{x},\mathbf{x}) - 2\sum_n \gamma_{i,n} \kappa(\mathbf{x},\mathbf{x}_n) + \sum_{n,m} \gamma_{m,n} \kappa(\mathbf{x}_n,\mathbf{x}_m) \tag{1}$$

The update of the mean is based on a soft learning algorithm,

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \zeta[\phi(\mathbf{x}) - \mathbf{m}_i(t)] \tag{2}$$

where ζ is the normalised winning frequency of the i-th mean. The above updating formula can be expressed in kernel function form as,

$$\gamma_{i,n}(t+1) = \begin{cases} \gamma_{i,n}(t)(1-\zeta), & \text{for } n \neq t+1 \\ \zeta, & \text{for } n = t+1 \end{cases} \tag{3}$$

Type II Kernel SOM

There is another, direct way to kernelise the SOM by mapping the data point to the feature space then applying the SOM in the mapped space. The winning rules of this second type of kernel SOM have been proposed as follows either in the input space [14],

$$v = \arg\min_i \|\mathbf{x} - \mathbf{m}_i\|, \tag{4}$$

or in the feature space [4]:

$$v = \arg\min_i \| \phi(\mathbf{x}) - \phi(\mathbf{m}_i) \| \tag{5}$$

The weight updating rule is proposed as [4, 14],

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t) h(v(\mathbf{x}), i) \nabla J(\mathbf{x}, \mathbf{m}_i) \tag{6}$$

where $J(\mathbf{x}, \mathbf{m}_i) = \| \phi(\mathbf{x}) - \phi(\mathbf{m}_i) \|^2$ is the distance function in the feature space or the proposed objective function. While, $\alpha(t)$ and $h(v(\mathbf{x}), i)$ are the learning rate and neighbourhood function respectively.

Note that,

$$J(\mathbf{x}, \mathbf{m}_i) = \| \phi(\mathbf{x}) - \phi(\mathbf{m}_i) \|^2 = \kappa(\mathbf{x}, \mathbf{x}) + \kappa(\mathbf{m}_i, \mathbf{m}_i) - 2\kappa(\mathbf{x}, \mathbf{m}_i) \tag{7}$$

and

$$\nabla J(\mathbf{x}, \mathbf{m}_i) = \frac{\partial \kappa(\mathbf{m}_i, \mathbf{m}_i)}{\partial \mathbf{m}_i} - 2 \frac{\partial \kappa(\mathbf{x}, \mathbf{m}_i)}{\partial \mathbf{m}_i} \tag{8}$$

Therefore this kernel SOM can be entirely operated in the feature space.

Link to Energy Function

From the energy function point of view, the SOM minimises the following energy [11, 8], at least for the discrete case,

$$E = \sum_i \int_{V_i} \sum_j h(i, j) \| \mathbf{x} - \mathbf{m}_j \|^2 \; p(\mathbf{x}) d\mathbf{x} \tag{9}$$

where V_i is the Voronoi tesselation of the neuron i.

The extension of this energy function in the feature space is,

$$E_F = \sum_i \int_{V_i} \sum_j h(i, j) \| \phi(\mathbf{x}) - \phi(\mathbf{m}_j) \|^2 \; p(\mathbf{x}) d\mathbf{x} \tag{10}$$

The kernel SOM can be seen as a result of direct minimising this transformed energy stochastically, i.e., using the sample gradient,

$$\frac{\partial \widehat{E}_F}{\partial \mathbf{m}_i} = \sum_j h(v(\mathbf{x}), j) \| \phi(\mathbf{x}) - \phi(\mathbf{m}_j) \|^2 = -2h(v(\mathbf{x}), i) \nabla J(\mathbf{x}, \mathbf{m}_j) \tag{11}$$

This leads to the weight updating rule Eq. (6).

Various kernel functions such as Gaussian, Cauchy, logarithm, polynomial, are readily applicable to the kernel SOM [12]. For example, for Gaussian kernel, the winning and weight updating rules are,

$$v = \arg\min_i J(\mathbf{x}, \mathbf{m}_i) = \arg\min_i [-2\kappa(\mathbf{x}, \mathbf{m}_i)] = \arg\min_i [-\exp(-\frac{\| \mathbf{x} - \mathbf{m}_i \|^2}{2\sigma^2})] \tag{12}$$

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t) h(v(\mathbf{x}), i) \frac{1}{2\sigma^2} \exp(-\frac{\| \mathbf{x} - \mathbf{m}_i \|^2}{2\sigma^2})(\mathbf{x} - \mathbf{m}_i) \tag{13}$$

respectively.

Experimental results on various benchmark datasets have shown that although kernel SOM does produce better classification performance when the kernel parameters are optimised (often empirically) in some cases and there is short of evidence to indicate that the kernel SOM will always outperform the SOM [12]. A typical set of results is given in Table 1.

Table 1: Classification errors on UCI colon cancer dataset. M, A and V denote the minimum distance, average distance and majority voting methods to label the nodes [12].

Kernel	*SOM*			*Type I Kernel SOM*			*Type II Kernel SOM*		
	M	A	V	M	A	V	M	A	V
Gaussian	4.3	7.0	3.8	5.6	5.8	5.6	5.3	5.3	4.7
Cauchy	3.8	7.5	3.7	5.5	5.6	5.5	5.5	5.5	4.8
Log	4.4	7.2	4.1	4.6	4.6	4.6	5.2	5.2	4.6

3 Self-Organising Mixture Network

The self-organising mixture network (SOMN) [18] extends and adapts the SOM to a mixture density model, in which each node characterises a conditional probability distribution. The joint-probability density of the data (or the network) is described by a mixture distribution,

$$p(\mathbf{x}\,|\,\Theta)=\sum_{i=1}^{K}p_i(\mathbf{x}\,|\,\theta_i)P_i \tag{14}$$

where $p_i(\mathbf{x}\,|\,\theta_i)$ is the i-th component-conditional density, and θ_i is the parameter for the i-th conditional density, i=1, 2, ... K, $\Theta=(\theta_1,\theta_2,...\theta_K)^T$, and P_i is the prior probability of the i-th component or node and is also called the mixing weights. For example, a Gaussian or Cauchy mixture has the following the conditional densities respectively,

$$p_i(\mathbf{x}\,|\,\theta_i)=\frac{1}{(2\pi)^{d/2}\,|\Sigma_i\,|^{1/2}}\exp[-\frac{1}{2}(\mathbf{x}-\mathbf{m}_i)^T\Sigma_i^{-1}(\mathbf{x}-\mathbf{m}_i)] \tag{15}$$

$$p_i(\mathbf{x}\,|\,\theta_i)=\frac{1}{\pi\,|\Sigma_i\,|^{1/2}\,[1+(\mathbf{x}-\mathbf{m}_i)^T\Sigma_i^{-1}(\mathbf{x}-\mathbf{m}_i)]} \tag{16}$$

where $\theta_i=\{\mathbf{m}_i,\Sigma_i\}$ are the mean vector and covariance matrix respectively.

Suppose that the true environmental data density function and the estimated one are $p(\mathbf{x})$ and $\hat{p}(\mathbf{x})$ respectively. The Kullback-Leibler information metric measures the divergence between these two, and is defined as:

$$\mathbf{I}=-\int\log\frac{\hat{p}(\mathbf{x})}{p(\mathbf{x})}p(\mathbf{x})d\mathbf{x} \tag{17}$$

When the estimated density is modelled as a mixture distribution, i.e. a function of various sub-densities and their parameters, one can seek the optimal estimate of these parameters by minimising the Kullback-Leibler metric via its partial differentials in respect to every model parameter, i.e.

$$\frac{\partial \mathbf{I}}{\partial \theta_i} = -\int [\frac{1}{\hat{p}(\mathbf{x}\,|\,\hat{\Theta})}\frac{\partial \hat{p}(\mathbf{x}\,|\,\hat{\Theta})}{\partial \theta_i}] p(\mathbf{x}) d\mathbf{x}\,, \qquad i=1, 2,...K. \tag{18}$$

As the true data density is unknown, the stochastic gradient was used for solving these non-directly solvable equations. This results in the following adaptive updating rules for the parameters and priors [18],

$$\begin{aligned} \hat{\theta}_i(t+1) &= \hat{\theta}_i(t) + \alpha(t) h(v(\mathbf{x}), i)[\frac{1}{\hat{p}(\mathbf{x}\,|\,\hat{\Theta})}\frac{\partial \hat{p}(\mathbf{x}\,|\,\hat{\Theta})}{\partial \theta_i}] \\ &= \hat{\theta}_i(t) + \alpha(t) h(v(\mathbf{x}), i)[\frac{\hat{P}_i(t)}{\sum_j \hat{P}_i(t)\hat{p}_j(\mathbf{x}\,|\,\theta_j)}\frac{\partial \hat{p}_i(\mathbf{x}\,|\,\hat{\theta}_i)}{\partial \theta_i}] \end{aligned} \tag{19}$$

$$\begin{aligned} \hat{P}_i(t+1) &= \hat{P}_i(t) + \alpha(t)[\frac{\hat{p}_i(\mathbf{x}\,|\,\hat{\theta}_i)\hat{P}_i(t)}{\hat{p}(\mathbf{x}\,|\,\hat{\Theta})} - \hat{P}_i(t)] \\ &= \hat{P}_i(t) - \alpha(t) h(v(\mathbf{x}), i)[\hat{P}(i\,|\,\mathbf{x}) - \hat{P}_i(t)] \end{aligned} \tag{20}$$

where α(t) is the learning coefficient or rate at time step t, and 0<α(t)<1 and decreases monotonically. The neighbourhood function $h(v(\mathbf{x}), i)$ is further introduced to restrict the learning in a neighbourhood of the winner, which is found via maximum (estimated) posterior probability of the node,

$$\hat{P}(i\,|\,\mathbf{x}) = \frac{\hat{P}_i \hat{p}_i(\mathbf{x}\,|\,\hat{\theta}_i)}{\hat{p}(\mathbf{x}\,|\,\hat{\Theta})} \tag{21}$$

When the SOMN is limited to the homoscedastic case, i.e. equal variances and equal priors (or non-informative priors) for all components, only the means are the learning variables. The above winner rule becomes,

$$v = \arg\max_i \frac{\hat{p}_i(\mathbf{x}\,|\,\theta_i)}{\sum_j \hat{p}_i(\mathbf{x}\,|\,\theta_j)} \tag{22}$$

Eq. (22) is equivalent to rule Eq. (5) or (7), when the density function is isotopic or a function of |||**x**-**m**||.

The corresponding weight updating is,

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t) h(v(\mathbf{x}), i) \frac{1}{\sum_j p_j(\mathbf{x}\,|\,\theta_j)} \frac{\partial p_i(\mathbf{x}\,|\,\theta_i)}{\partial \mathbf{m}_i} \tag{23}$$

It can be seen that Eq. (23) bears similarity to Eq. (6). Again if the conditional density function is of a kernel type and isotopic or vice versa (e.g. Gaussian and Cauchy functions), the above rule leads to the same result as Eq. (6), with an additional normalising factor $\sum_j \hat{p}_i(\mathbf{x}\,|\,\theta_j) = p(\mathbf{x})$. When the data density is relatively smooth, this factor is only a (same) scalar value to all nodes.

For example, for a Gaussian mixture with equal variance and prior for all nodes, it is easy to show that the winning and mean updating rules are,

$$v = \arg\max_i [\exp(-\frac{\|\mathbf{x} - \mathbf{m}_i\|^2}{2\sigma^2})] \tag{24}$$

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t)h(v(\mathbf{x}),i)\frac{1}{2\sigma^2}\frac{1}{\sum_j p_j(\mathbf{x}\,|\,\theta_j)}\exp(-\frac{\|\mathbf{x}-\mathbf{m}_i\|^2}{2\sigma^2})(\mathbf{x}-\mathbf{m}_i) \tag{25}$$

They are equivalent to those of the kernel SOM with Gaussian kernels, i.e., Eqs.(12) and (13).

The equivalence between the SOMN and kernel SOM on one hand explains that the kernel SOM is approximating a mixture density model using the kernel function as the prototype conditional density. On the other hand, as the SOM is a special case of the SOMN with equal variance and prior for all nodes and when the number of nodes is great, the SOM is natural kernel method.

4 Conclusions and Discussions

In this paper, the relation between kernel SOM and self-organising mixture network (SOMN) has been established. When the conditional density function is of a kernel type or the kernel function is of density type, and both are isotopic function, then two methods are equivalent. Then the kernel SOM can be understood as an entropy optimised mixture density learner. As the SOM is a special case of SOMN, this in turn explains that the SOM approximates the kernel method naturally.

References

[1] M. Aizerman, E. Braverman and L. Rozonoer (1964), Theoretical foundations of the potential function method in pattern recognition learning, *Automation and Remote Control*, vol. 25, 821-837.

[2] N. Allinson, K. Obermayer, H. Yin (Eds.) (2002), *Special Issue on New Developments on Self-Organising Maps, Neural Networks*, vol. 15.

[3] N. Allinson, H. Yin, L. Allinson and J. Slack (Eds.) (2001), *Advances in Self-Organising Maps,* London, Springer.

[4] P. Andras (2002), Kernel-Kohonen networks, *International Journal of Neural Systems*, vol 12, 117-135.

[5] C. Cortes and V. Vapnik (1995), Support vector networks, *Machine Learning*, vol. 20, 273-297.

[6] C. Fyfe, D. MacDonald and D. Charles (2000), Unsupervised learning using radial kernels, in: Howlett, eds. *Recent Advances in Radial Basis Networks.*

[7] R.J. T. Graepel, M. Burger and K. Obermayer (1998), Self-organizing maps: Generalization and new optimization techniques, *Neurocomputing*, vol 21, 173-190.

[8] T. Heskes (1999), Energy functions for self-organizing maps, In E. Oja and S. Kaski, eds, *Kohonen Maps*, Elsevier.

[9] M. Ishikawa, R. Miikulainen and H. Ritter (Eds.) (2004), *Special Issue on New Developments on Self-Organising Systems, Neural Networks,* vol. 17.

[10] T. Kohonen (1999), Self-Organising Maps, Springer.

[11] J. Lampinen and E. Oja (1992), Clustering properties of hierarchical self-organizing maps, *Journal of Mathematical Imaging and Vision*, vol 2, 261-272.

[12] K.W. Lau and H. Yin (2005), Kernel self-organising maps for classification, submitted to *Neurocomputing*.

[13] D. MacDonald and C. Fyfe (2000), The kernel self organising map, Applied Computational Intelligence Research Unit, The University of Paisley.

[14] Z.S. Pan, S.C. Chen and D.Q. Zhang (2004), A kernel-base SOM classifer in input space, *Acta Electronica Sinica*, vol 32, 227-231 (in Chinese).

[15] B. Schölkopf, A. Smola and K.R. Müller (1998), Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation*, vol. 10, 1299-1319.

[16] J. Shawe-Taylor and N. Cristianini (2004), *Kernel Methods for Pattern Analysis*, Cambridge University Press.

[17] M. van Hulle (2002), Kernel-based topographic map formation achieved with an information-theoretic approach, *Neural Networks*, vol 15, 1029-1039.

[18] H. Yin and N. Allinson (2001), Self-organising mixture networks for probability density estimation, *IEEE trans. Neural Networks*, vol 12, 405-411.