ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

Kernel Generative Topographic Mapping

Iván Olier[1], Alfredo Vellido[2] and Jesús Giraldo[3,1] *

1- Institut de Neurosciènces, and 3- Unitat de Bioestadística
Universitat Autònoma de Barcelona
Edifici M - CP 08193, Bellaterra (Barcelona) - Spain

2- Departament de Llenguatges i Sistemes Informatics
Universitat Politècnica de Catalunya
C/. Jordi Girona 1-3, Edifici Omega, CP 08034, Barcelona - Spain

Abstract. A kernel version of Generative Topographic Mapping, a model of the manifold learning family, is defined in this paper. Its ability to adequately model non-i.i.d. data is illustrated in a problem concerning the identification of protein subfamilies from protein sequences.

1 Introduction

Manifold learning models attempt to describe multivariate data in terms of low dimensional representations, usually in order to achieve an intuitive visualization of high dimensional data. Visualization may help in the exploratory stages of data analysis. Generative Topographic Mapping (GTM) [1], whose probabilistic setting and functional similarities make it a principled alternative to Self-Organizing Maps (SOM) [2], is a model of this family defined for the clustering and visualization of i.i.d. data. Although several variants have been developed for various types of data (e.g., [3, 4]), GTM lacks the ability to handle more structured data, such as strings, trees, or graphs.

Kernelization is a method originally defined for Support Vector Machines (SVM). It has been pointed out that it could be used to develop generalizations of any algorithm that could be cast in dot product terms. Recent years have witnessed the development of models such as Kernel Principal Components Analysis (KPCA) [5], Kernel Fisher Discriminant Analysis (KFDA) [6], or kernel SOM [7], amongst others. The idea is that a method formulated in terms of kernels can use the one that best suits the problem and data type at hand. With this purpose, we define kernel-GTM (KGTM). It takes advantage of the original GTM functionalities to achieve clustering and visualization of a wider variety of data types. The capabilities of KGTM are first illustrated through experimentation with artificial data. The model is then applied to a problem concerning the clustering and visualization of protein sequences.

*This research was partially supported by Catalan La Marató de TV3 Foundation project 070530 and Spanish MICINN projects TIN2009-13895-C02-01 and SAF2007-65913.

ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

2 Background

2.1 The Original GTM

The neural network-inspired GTM is a nonlinear latent variable model of the manifold learning family, with sound foundations in probability theory. It performs simultaneous clustering and visualization of the observed data through a nonlinear and topology-preserving mapping from a visualization latent space in $\Re^\ell$ (with ℓ being usually 1 or 2 for visualization purposes) onto a manifold embedded in the $\Re^D$ space, where the observed data reside. The mapping that generates the manifold is carried out through a generalized regression function:

$$\mathbf{y} = \mathbf{W}\boldsymbol{\phi}(\mathbf{u}) \tag{1}$$

where $\mathbf{y} \in \Re^D$, $\mathbf{u} \in \Re^\ell$, $\mathbf{W}$ is the matrix that generates the mapping, and $\boldsymbol{\phi}$ is a vector with the images of S basis functions ϕ_s To achieve computational tractability, the prior distribution of $\mathbf{u}$ in latent space is constrained to form a uniform discrete grid of M centres, analogous to the layout of the SOM units, in the form of a sum of delta functions $\mathbf{u} = \frac{1}{M}\sum_{m=1}^{M} \delta\mathbf{u} - \mathbf{u}_m$.

This way defined, the GTM can also be understood as a special case of a Gaussian mixture model that is adapted to provide high-dimensional data visualization. Each component m in the mixture defines the probability of an observable data point $\mathbf{x}$ given a latent point $\mathbf{u}_m$ and model:

$$p(\mathbf{x}|\mathbf{u}_m, \boldsymbol{\Theta}) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2}\left\|\mathbf{x} - \mathbf{y}_m\right\|^2\right\} \tag{2}$$

where $\mathbf{y}_m = \mathbf{W}\boldsymbol{\phi}(\mathbf{u}_m)$. The set of adaptive parameters $\boldsymbol{\Theta}$ is constituted by $\mathbf{W}$ and the common inverse variance β. A density model in data space is therefore generated for each component m of the mixture, which, assuming that the observed data set $\mathbf{X}$ consists of N independent, identically distributed (i.i.d.) data points $\mathbf{x}_n$, leads to the definition of a likelihood in the form:

$$\mathcal{L}(\mathbf{W}, \beta) = \prod_{n=1}^{N} \frac{1}{M} \sum_{m=1}^{M} p(\mathbf{x}_n|\mathbf{u}_m, \mathbf{W}, \beta) \tag{3}$$

The adaptive parameters of the model are usually optimized by Maximum Likelihood (ML) using the Expectation-Maximization (EM) algorithm [8]. Details can be found in [1].

2.2 The *kernel trick*

Kernelization was originally devised for SVM. The idea is that observed data $\mathbf{X}$ can be implicitly mapped into a high-dimensional feature space H via a nonlinear function: $\mathbf{x} \to \psi(\mathbf{x})$. A similarity measure can then be defined from the dot product in space H as follows:

$$K(\mathbf{x}, \mathbf{x}') = \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle \tag{4}$$

K is a kernel function that should satisfy Mercer's condition [9]. It allows us to deal with learning algorithms using linear algebra and analytic geometry. In general, this method deals with data in the high-dimensional dot product space H, usually known as feature space. This use of the feature space avoids expensive computation costs by employing the kernel function K instead of directly computing the dot product in H.

3 Kernel Generative Topographic Mapping

The kernelization of GTM can be implemented by redefining Eq. 2 in feature space as

$$p\left(\boldsymbol{\psi}\left(\mathbf{x}\right)|\mathbf{u}_m, \boldsymbol{\Theta}\right) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2}\left\|\boldsymbol{\psi}\left(\mathbf{x}\right) - \mathbf{y}_m\right\|^2\right\} \tag{5}$$

Note that the prototypes $\mathbf{y}_m$ are now defined in the feature space and not in data space, as originally. Consequently, D is now the dimension of the feature space, which is usually unknown. In most cases, the term $\left\|\boldsymbol{\psi}\left(\mathbf{x}\right) - \mathbf{y}_m\right\|^2$ cannot be directly evaluated, given that the function $\psi\left(\cdot\right)$ is usually unknown. However, this term can be also expressed as follows:

$$\left\|\boldsymbol{\psi}\left(\mathbf{x}\right) - \mathbf{y}_m\right\|^2 = \left\langle\boldsymbol{\psi}\left(\mathbf{x}\right), \boldsymbol{\psi}\left(\mathbf{x}\right)\right\rangle + \left\langle\mathbf{y}_m, \mathbf{y}_m\right\rangle - 2\left\langle\boldsymbol{\psi}\left(\mathbf{x}\right), \mathbf{y}_m\right\rangle \tag{6}$$

Here, we assume that, as in KPCA, $\mathbf{y}_m$ can be expanded on the training data in the feature space. That is, $\mathbf{y}_m = \boldsymbol{\Psi}\boldsymbol{\omega}_m$, where $\boldsymbol{\Psi}$ is a $D \times N$-matrix of vector columns $\boldsymbol{\psi}\left(\mathbf{x}_n\right)$, $n = 1 \ldots N$, and $\boldsymbol{\omega}_m$ a weight vector. With the aim of preserving the topology, we correlate the weight vector to the latent space by $\boldsymbol{\omega}_m = \boldsymbol{\Lambda}\boldsymbol{\phi}_m$, where $\boldsymbol{\Lambda}$ is an adaptative weight matrix and $\boldsymbol{\phi}_m = \boldsymbol{\phi}\left(\mathbf{u}_m\right)$ is the set of radial basis functions typically used by GTM. Therefore, Eq. 6 becomes:

$$\left\|\boldsymbol{\psi}\left(\mathbf{x}_n\right) - \mathbf{y}_m\right\|^2 = J_{mn} = K_{nn} + \left(\boldsymbol{\Lambda}\boldsymbol{\phi}_m\right)^T \mathbf{K}\boldsymbol{\Lambda}\boldsymbol{\phi}_m - 2\mathbf{k}_n\boldsymbol{\Lambda}\boldsymbol{\phi}_m \tag{7}$$

where $\mathbf{K}$ is a kernel matrix with elements $K_{nn'} = \left\langle\boldsymbol{\psi}\left(\mathbf{x}_n\right), \boldsymbol{\psi}\left(\mathbf{x}_{n'}\right)\right\rangle$, and row vectors $\mathbf{k}_n$. Thereby J_{mn} is expressed in terms of the kernel matrix, making the definition of function $\boldsymbol{\psi}\left(\cdot\right)$ unnecessary. The adaptive parameters of the model are now $\boldsymbol{\Lambda}$ and β, which can be optimized by ML using EM, as in GTM. The likelihood of the model is formulated as follows:

$$\mathcal{L}\left(\boldsymbol{\Lambda}, \beta\right) = \prod_{n=1}^{N} \frac{1}{M} \sum_{m=1}^{M} p\left(\boldsymbol{\psi}\left(\mathbf{x}_n\right)|\mathbf{u}_m, \boldsymbol{\Lambda}, \beta\right) \tag{8}$$

Following the usual EM algorithm, the expectation step proceeds with the estimation of the posterior distribution $R_{mn} = p\left(\mathbf{u}_m|\boldsymbol{\psi}\left(\mathbf{x}_n\right), \boldsymbol{\Lambda}, \beta\right)$ as:

$$R_{mn} = \frac{p\left(\boldsymbol{\psi}\left(\mathbf{x}_n\right)|\mathbf{u}_m, \boldsymbol{\Lambda}, \beta\right)}{\sum_{m'=1}^{M} p\left(\boldsymbol{\psi}\left(\mathbf{x}_n\right)|\mathbf{u}_{m'}, \boldsymbol{\Lambda}, \beta\right)} \tag{9}$$

R_{mn} measures the degree of responsability of a point $\mathbf{u}_m$ in the latent space for the generation of a $\boldsymbol{\psi}(\mathbf{x}_n)$ point in the feature space. In turn, each R_{mn} is an element of a $M \times N$ *responsability matrix* $\mathbf{R}$.

In the maximization step we use Eq. 8 as the optimization function to determine the parameters $\boldsymbol{\Lambda}$ and β, which results in the following expressions:

$$\boldsymbol{\Lambda}^T = \left(\boldsymbol{\Phi}^T \mathbf{G} \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^T \mathbf{R} \tag{10}$$

$$\frac{1}{\beta} = \frac{1}{ND} \sum_{n=1}^{N} \sum_{m=1}^{M} R_{mn} J_{mn} \tag{11}$$

Starting with a random initialization of these parameters, steps E and M of EM are sequentially repeated until convergence of the likelihood function is reached.

4 Experiments

4.1 Artificial dataset

A first experiment was carried out to preliminary assess the differential ability of KGTM to faithfully represent and visualize data. For that, an artificial data set consisting of two spirals, as displayed in Figure 1(a), was generated. The projections of these data in the latent spaces of GTM and KGTM are, in turn, shown in Figure 1(b) and Figure 1(c). KGTM was implemented using a Gaussian kernel and it is shown to capture the inherent structure of this dataset far better than the standard GTM.

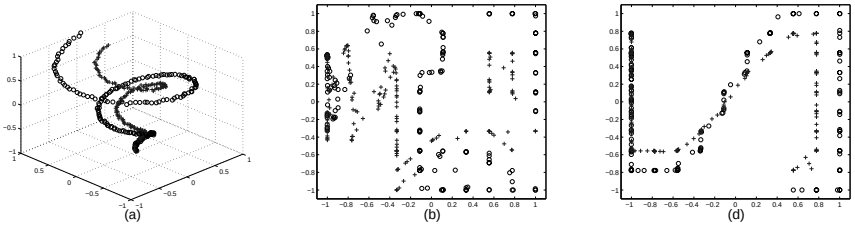

Fig. 1: (a) Plot of the artificial dataset; (b) projection of dataset produced by GTM using the *mean projection* of data: $\mathbf{u}_{mean} = \sum_{m=1}^{M} R_{mn} \mathbf{u}_m$; (c) mean projection of the data, now using the proposed KGTM. In all plots, each of the spirals is represented by different symbols: '+' and 'o'.

4.2 Protein subfamily visualization using protein sequences

Protein classification in families is a frequent problem in bioinformatics. To date, many protein sequences remain *orphans*, meaning that they do not belong to any particular family and thus its function is unknown. A particular type of proteins

ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

are the G-protein coupled receptors (GPCR) , traditionally divided into three big families (usually named A, B and C) and, in turn, into subfamilies. They represent nearly half of the current market for therapeutic agents and remain a primary focus of many biomedical research and drug discovery programs.

For illustration, we have designed an experiment to model the family C of GPRCs only, using KGTM to explore its subfamilies (7 in total) through visualization in the latent space. The dataset consists of 232 protein sequences obtained from GPCRDB[1]. Each position in a sequence is called a *residue*, which in turn may be one of 20 possible amino acids. Each amino acid has a standard one-letter code, thus a protein is represented by a sequence of these letters. The number of residues by protein sequence in the dataset is 253.

A key issue in this problem is the design of an appropriate kernel function for measuring protein sequences similarities. We designed a kernel funtion based on the mutations and gaps between the sequences, which takes the form:

$$K(x, x') = \rho \exp\left\{\nu \frac{\pi(x, x')}{\pi(x, x) + \pi(x', x')}\right\} \tag{12}$$

where x and x' are two sequences, and ρ and ν are prefixed parameters ; $\pi(\cdot)$ is a score function, commonly used in bioinformatics and expressed as follows: $\pi(x, x') = \sum_r s(x_r, x'_r) - \gamma$, where x_r and x'_r are the r^{th} residue in the sequences. The value of $s(x_r, x'_r)$ can be found in a mutation matrix [10] and γ is a gap penalty (usually the number of gaps in sequences). This kernel function could be seen as a symmetric version of asymmetric bio-basis functions [11], which are not suitable for applying Mercer's theorem.

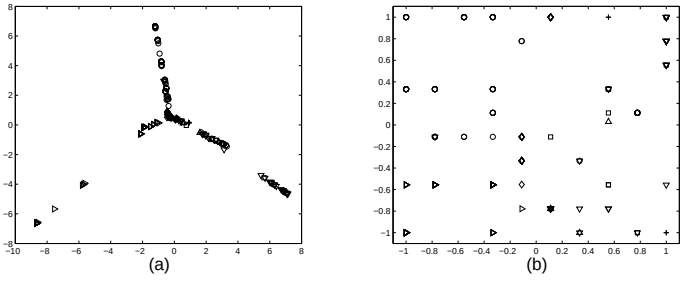

Fig. 2: Protein sequence visualization using (a) KPCA and (b) KGTM. Each of the GPCR subfamilies is coded in the map using different symbols.

Visualization results for KGTM are shown in Fig. 2 and compared to those obtained using the 1^{st} and 2^{nd} principal components of a Kernel PCA model. The map produced by KPCA does not capture the structure of the data in a way that allows us to differentiate between GPCR subfamilies. Instead, KGTM provides a far more clear separation between subfamilies, which can be better appreciated in the maps of Fig. 3. Here, the data are visualized in the latent

[1]GPCRDB web site: http://www.gpcr.org/7tm/

space using the *mode-projection*, which is defined as: $m_{mode} = \underset{m}{\texttt{argmax}}\ R_{mn}$. Each subfamily occupies a rather differentiated area on the map, showing little overlapping. Future research should qualify KGTM capabilities using a wider variety of artificial data. A more thorough quantitative analysis of the protein subfamily characterization problem should also be carried out.

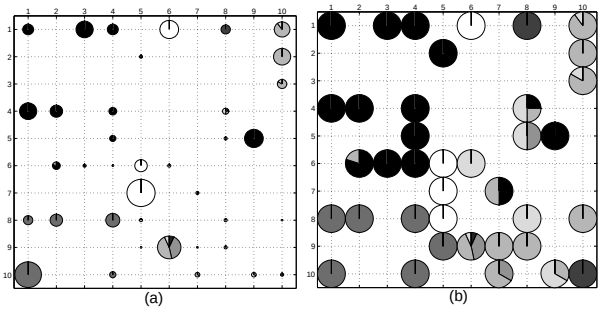

Fig. 3: Data visualization using a mode-projection. Left) Pie charts represent latent points, and their size is proportional to the ratio of sequences assigned to them. Each portion of the charts corresponds to the percentage of sequences belonging to each subfamily, which are color-coded in shades of gray. Right) Same map without scaling, for better visualization.

References

[1] C. M. Bishop, M. Svensén, and C. K. I. Williams, GTM: The Generative Topographic Mapping, *Neural Comput.*, 10(1):215–234, Elsevier, 1998.

[2] T. Kohonen. *Self-Organizing Maps (3rd ed).* Springer-Verlag, Berlin, 2001.

[3] I. Olier and A. Vellido. Advances in clustering and visualization of time series using GTM Through Time. *Neural Networks*, 21(7):904–913, Elsevier, 2008.

[4] M. Girolami, Latent variable models for the topographic organisation of discrete and strictly positive data, *Neurocomp.*, 48:185–198, Elsevier, 2002.

[5] B. Schölkopf, A. Smola, and K. R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, *Neural Comput.*, 10(5):1299–1319, MIT Press, 1998.

[6] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müller. Fisher discriminant analysis with kernels. In *IEEE Workshop on Neural Networks for Signal Processing IX*, pages 41–48, 1999.

[7] N. Villa and F. Rossi, A comparison between dissimilarity SOM and kernel SOM for clustering the vertices of a graph. In *proceedings of the 6th Workshop on Self-Organizing Maps* (WSOM 07), Bielefield, Germany, 2007.

[8] A. P. Dempster, M. N. Laird and D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *J. Roy. Stat. Soc. B*, 39:1–38, 1977.

[9] B. Schölkopf and A. Smola. *Learning with Kernels.* The MIT Press, Cambridge, Massachussets, 2002.

[10] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids.* Cambridge Univ. Press, Cambridge, 2004.

[11] Z. R. Yang and R. Thomson, A novel neural network method in mining molecular sequence data, *IEEE T. Neural Networ.*, 16:263–274, 2005.