

Constrained Mixture Modeling of Intrinsically Low-Dimensional Distributions

Joris Portegies Zwart

IAS Group, Dept. of Computer Science
University of Amsterdam, The Netherlands
portegie@science.uva.nl

TNO Physics and Electronics Laboratory
The Hague, The Netherlands

Ben Kröse

IAS Group, Dept. of Computer Science
University of Amsterdam, The Netherlands
krose@science.uva.nl

Abstract

In this paper we introduce a novel way of modeling distributions with a low latent dimensionality. Our method allows for a strict control of the properties of the mapping between the latent and the feature space. Usually, as in for example GTM, this mapping is constructed through the maximization of the log likelihood of the data set. However, if the data set is supervised, in the sense that we know the corresponding latent vector value for each feature vector, it is more sensible to use some regression method for finding the mapping in advance. The mapping is then fixed during optimization of the log likelihood of the data set.

It is concluded that in terms of log likelihood the methods are comparable. The advantages however lie in the better understanding of the properties of the mapping and a clear interpretation of the latent variables.

1. Introduction

One possible approach for automatic pattern recognition is through the use of probability density functions (pdf). Given the class-conditional pdf $p(\mathbf{x}|C_t)$ for each target class C_t , classification follows through the use of Bayes' rule,

$$p(C_t|\mathbf{x}) = \frac{p(\mathbf{x}|C_t)p(C_t)}{p(\mathbf{x})}. \quad (1)$$

Each test vector \mathbf{x} is then assigned the class label C_t for which $p(C_t|\mathbf{x})$ is maximal.

Usually the true density $p(\mathbf{x}|C_t)$ is not known, and consequently has to be estimated from the available training data. In this paper we focus on the case where the distribution is governed by a low-dimensional parameter, i.e. a latent variable.

In many classification applications we have to deal with high dimensional data $\{\mathbf{x}_n\} \in V$ generated by a process which possesses only a few degrees of freedom. We would like to model the distribution $p(\mathbf{x})$ in V parametrized by intrinsic variables $\boldsymbol{\theta}$ in a low dimensional vector space M .

If we have knowledge of both \mathbf{x}_n and the corresponding $\boldsymbol{\theta}_n$ during training this knowledge can be used to model $p(\mathbf{x}|\boldsymbol{\theta})$ [1] by optimizing the log likelihood of the training set $\{\mathbf{x}_n, \boldsymbol{\theta}_n\}$. This distribution can be used during classification provided that $\boldsymbol{\theta}$ is known then as well.

If on the other hand, we know nothing about the underlying $\boldsymbol{\theta}$'s, we can only guess the relationship between \mathbf{x} and $\boldsymbol{\theta}$, using $\boldsymbol{\theta}$ as a dummy variable whose sole purpose is enforcing the dimensionality restriction. After estimating $p(\mathbf{x}|\boldsymbol{\theta})$, the dependence on $\boldsymbol{\theta}$ has to be integrated out to obtain $p(\mathbf{x})$,

$$p(\mathbf{x}) = \int p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}. \quad (2)$$

In the neural network community, the Generative Topographic Mapping (GTM) [2] is one of the more popular methods for dealing with this situation. It is a mixture model [3], which means equation (2) is approximated by a sum over K kernels,

$$p(\mathbf{x}) = \sum_{k=1}^K p(\mathbf{x}|\boldsymbol{\theta}_k) p_k, \quad (3)$$

in which $p_k = p(\boldsymbol{\theta}_k)d\boldsymbol{\theta}$. Usually the prior distribution over $\boldsymbol{\theta}$ is assumed to be uniform, $p_k = 1/K$. In GTM, the kernels are multivariate Gaussians \mathcal{N} with variance σ^2 , whose centers $\boldsymbol{\mu}_k$ are restricted to a manifold defined by a mapping $F : M \rightarrow V : \boldsymbol{\theta} \rightarrow \mathbf{x}$, so $\boldsymbol{\mu}_k = F(\boldsymbol{\theta}_k)$. Then equation (3) can be written as

$$p(\mathbf{x}) = \sum_{k=1}^K \mathcal{N}(\mathbf{x}|F, \boldsymbol{\theta}_k, \sigma) p_k. \quad (4)$$

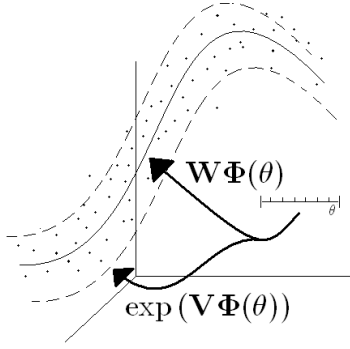


Figure 1. Fully supervised modeling. Both the mean $\mu(\theta)$ and the standard deviation $\sigma(\theta)$ are modeled as radial basis functions, such that the resulting likelihood of the data set $\{\mathbf{x}_n, \theta_n\}$ is maximized.

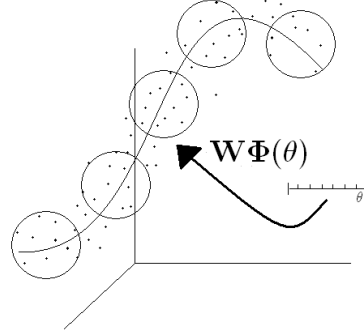


Figure 2. Schematic representation of the GTM procedure. A total of K θ_k 's are chosen uniformly in an interval $[-1, 1]$. The distribution $p(\mathbf{x})$ is describes as a sum over K spherical Gaussian kernels with variance σ^2 and centers $\mu(\theta_j, W)$.

The mapping F is constructed by optimizing the resulting log likelihood of the training set $\{\mathbf{x}_n\}$.

A third possibility is that the training set consists of supervised data $\{\mathbf{x}_n, \theta_n\}$, but during classification no information regarding θ is available. The question then arises how this information can be used for estimating $p(\mathbf{x})$. In this paper we describe a new method for supervised modeling of high dimensional data with a low latent dimensionality which deals with this situation. We start with a short introduction to fully supervised modeling (section 2) and GTM (section 3). Our method, described in section 4 is a mixture model like GTM, but unlike GTM is trained using a supervised data set. We investigate whether this model gives a better description of the data $\{\mathbf{x}_n\}$.

2. Fully Supervised Modeling

Consider both μ and σ as functions of θ , using a Radial Basis Function (RBF) approximation [2, 4, 1]:

$$\mu(\theta) = \mathbf{W}\phi(\theta), \quad (5)$$

$$\sigma(\theta) = \exp(\mathbf{V}\phi(\theta)). \quad (6)$$

The conditional distribution of the data can now be written as $p(\mathbf{x}|\theta, \mathbf{W}, \mathbf{V})$ (see figure 1). The log-likelihood \mathcal{L} is given by

$$\mathcal{L} = \log \prod_{n=1}^N p(\mathbf{x}_n|\theta_n, \mathbf{W}, \mathbf{V}) \quad (7)$$

$$= \sum_{n=1}^N \log \mathcal{N}(\mathbf{x}_n|\mu(\theta_n), \sigma(\theta_n)). \quad (8)$$

According to [1] this may be optimized using some gradient scheme, but no explicit algorithm is given. This approach is closely related to the work presented in [5, 4] on regression with input-dependent noise.

Optimization of the likelihood is achieved through an iterative two-level procedure [5, 4]. First keep \mathbf{V} fixed and minimize with respect to \mathbf{W} , then keep \mathbf{W} fixed and minimize with respect to \mathbf{V} .

3. GTM

In the GTM [2] method, the centers $\mu(\theta_k)$ of the kernels and their variance σ^2 (the same for all kernels) are adjusted such that the likelihood of the training set $\{\mathbf{x}_n\}$ is maximized (see figure 2). The positions of the centers depend on the mapping $\theta \rightarrow \mu(\theta)$ and on the choice of θ_k . In GTM the latter is fixed. The mapping from θ to μ is modeled with a RBF:

$$\mu = \mu(\theta) = \mathbf{W}\phi(\theta). \quad (9)$$

The choice of the number of basis functions determines the 'smoothness' of the curve on which the centers are positioned. The mixture model can now be written as:

$$p(\mathbf{x}) = \sum_{k=1}^K p(\mathbf{x}|\mathbf{W}, \sigma) p_k, \quad (10)$$

and the log likelihood of a data set $\{\mathbf{x}_n\}$ is given by:

$$\mathcal{L} = \sum_{n=1}^N \log \mathcal{N}(\mathbf{x}_n|\mathbf{W}, \sigma). \quad (11)$$

This log likelihood is then optimized in terms of \mathbf{W} and σ using an Expectation-Maximization (EM) algorithm.

4. Our method

So far, we have described two cases: either both \mathbf{x} and θ are known and so we're interested in $p(\mathbf{x}|\theta)$ (section 2), or we have no knowledge of θ and use GTM to implement the notion of intrinsic dimensionality and find $p(\mathbf{x})$ (section 3).

In our application (aircraft recognition using radar), we do have knowledge of both \mathbf{x} and θ during training, but during actual classification we only have \mathbf{x} . Our aim is to model $p(\mathbf{x})$ as a mixture model as in GTM, while using the information about θ for a better estimation of the kernel positions. It has been noted [6] that in GTM the topological ordering of the kernels does not necessarily correspond to the ordering in the latent space. Our approach doesn't suffer from this problem.

The new method we propose is similar to GTM in that the kernels of the mixture are required to lie on a low dimensional manifold in feature space defined by a mapping from θ to \mathbf{x} . The difference is that in our approach, this manifold is constructed at the start of the algorithm, and is kept fixed during further optimization steps. Also we allow for a separate σ for each kernel. We accomplish this as follows.

First, we construct the mapping F ,

$$F : M \rightarrow V : \theta \rightarrow F(\theta), \quad (12)$$

using a regression method. The exact functional form of this mapping is unimportant. If prior knowledge is available about the properties of the manifold (for instance, one could demand the manifold to be linear), this can be incorporated into the regression. Otherwise, a general regressor can be used. If gradient methods will be used for later optimization of the log likelihood (through equation (14), the only requirement is that the derivative $F'(\theta)$ has to be defined for all θ .

Now, we want to model $p(x)$ as a sum over K kernels, as in equation (3). However, instead of updating the centers of each kernel in high dimensional space (as in normal mixture models), or updating the parameters of the mapping (as in GTM) we update the location of the centers in the low-dimensional manifold, i.e. we try to find the optimal θ_k 's. This ensures that the kernel centers lie on the previously constructed manifold in the high dimensional space.

Finally, denoting the inverse variance of each kernel by $\beta_k = 1/\sigma_k^2$, the log likelihood is given by

$$L = \sum_{n=1}^N \log \sum_{k=1}^K p(\mathbf{x}_n | \theta_k, \beta_k) \quad (13)$$

with

$$p(\mathbf{x}_n | \theta, \beta) = \left(\frac{\beta}{2\pi} \right)^{D/2} \exp \left(-\frac{\beta}{2} \|F(\theta) - \mathbf{x}_n\|^2 \right).$$

Because in general F is non-invertible, an EM-type update equation for the latent kernel centers cannot be found, and we have to employ some non-linear optimization method. Gradient methods can be used, with the help of

$$\frac{\partial L}{\partial \theta_k} = - \sum_{n=1}^N \beta_k^{-1} [(F(\theta_k) - \mathbf{x}_n) \cdot F'(\theta_k)] p(\theta_k | \mathbf{x}_n). \quad (14)$$

The update equation for β_k is given by

$$\beta_k^{-1} = \frac{1}{D} \frac{\sum_n \|F(\theta_k) - \mathbf{x}_n\|^2 p(\theta_k | \mathbf{x}_n)}{\sum_n p(\theta_k | \mathbf{x}_n)}, \quad (15)$$

where D is the dimensionality of the feature space.

This method has a few advantages. First of all, it allows control over the mapping, and thus it allows for the definition of properties of the manifold (like, for instance, its shape). Furthermore, the latent variable θ is now 'physical' in the sense that its interpretation is clear.

If, however, we have no prior knowledge of our manifold, the regression step can lead to problems. When using a general regression method, we have to make a choice regarding the smoothness of the fit. In this context this implies we implicitly choose a noise level as well. In cases where we have no knowledge about the noise, this can lead to overfitting.

5. Experimental Results

To illustrate the difference between GTM and our new method described above, we run them both on a toy data set. This data set consists of 100 2D points generated by adding Gaussian noise to points randomly selected from a 1D manifold given by

$$\mathbf{x} = 10 [\sin(\pi t) \quad t + \cos(3\pi t)]^T, \quad (16)$$

with $t = [0, 1]$.

We trained all four methods on the same data set and report the resulting log likelihood of the training set. The results are given in figures 3. In each plot the data set is shown, together with the 1D manifold given by the final mapping.

In figure 3, the plot on the left shows the results from GTM, the plot on the right the results from our kernel method. Both distributions were modeled using 10 kernels. Obviously, the main difference is the shape of the resulting manifold. Since GTM is only concerned with the locations of the kernels, the shape of the manifold does not reflect the original ordering of our data set. Our method results in comparable locations for the kernels, while however preserving the original manifold. Since this manifold is the result of a normal regression, its shape now reflects the ordering of our

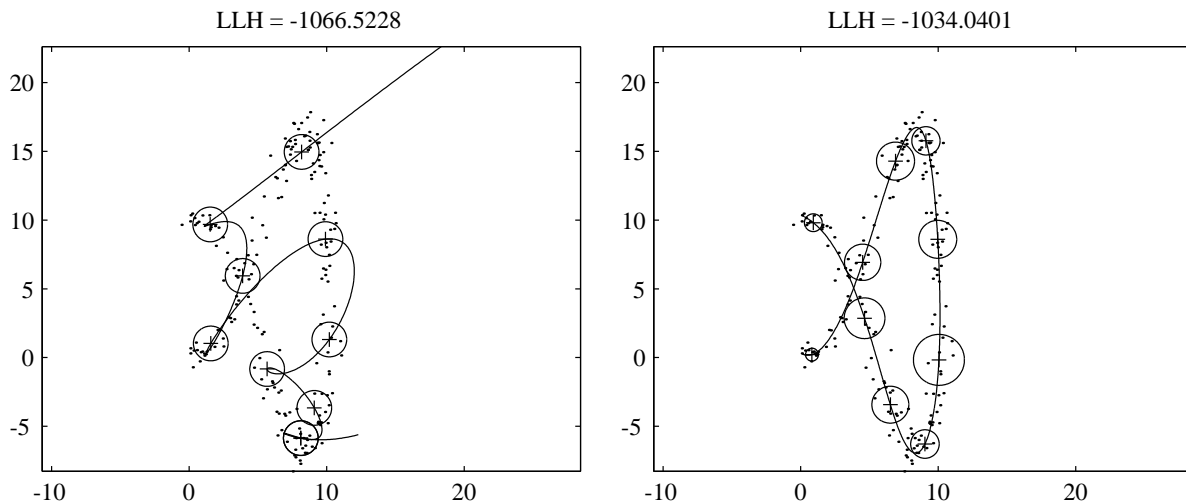


Figure 3. Location and width of kernels after modeling, GTM on the left, our new method on the right. Above each figure the resulting log likelihood for the training set is given.

data set. Furthermore, the kernel widths (specifically at the endpoints of the manifold) indicate a distribution which is peaked around the manifold.

6. Conclusion

In this paper we compared two different ways of modeling the distribution of a high dimensional data set with a low latent dimensionality. We proposed a new method, similar to GTM. In terms of log likelihood, both GTM and our new kernel method perform comparably. However, as can be seen in figure 3, our method results in a distribution which is more localized around the underlying manifold. This means that points *outside* the distribution will in general have a lower log likelihood using our method than using GTM.

7. Acknowledgements

We would like to thank Sjoerd Gelsema (TNO Physics and Electronics Laboratory), René van der Heiden (NATO C3 Agency) and Frans Groen and Nikos Vlassis (University of Amsterdam) for their advice.

References

[1] A.R. Webb. Gamma mixture models for target recognition. In *NATO RTO-SCI Symposium on Non-Cooperative Air Target Identification Using Radar*, Mannheim, 1998. ISBN: 92-837-0000-7.

- [2] C.M. Bishop, M. Svensén, and C.K.I. Williams. GTM: A principled alternative to the self-organizing map. In C. Von der Malsburg, W. Von Seelen, J.C. Vorbrüggen, and B. Sendhoff, editors, *Proceedings 1996 International Conference on Artificial Neural Networks*, pages 164–170. Springer-Verlag, 1997.
- [3] D. M. Titterton, A. F. Smith, and U. E. Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley, 1985.
- [4] Nikos Vlassis and Ben Kröse. Mixture conditional density estimation with the EM algorithm. In *Proc. ICANN'99*, 1999.
- [5] C.M. Bishop and C.S. Qazaz. Regression with input-dependent noise: A bayesian treatment. *Advances in Neural Information Processing Systems*, 9, 1997.
- [6] Kimmo Kiviluoto and Erkki Oja. S-map: A network with a simple self-organization algorithm for generative topographic mappings. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.