Manifold Constrained Finite Gaussian Mixtures

Cédric Archambeau* and Michel Verleysen**

Machine Learning Group - Université catholique de Louvain,
Place du Levant 3, B-1348 Louvain-la-Neuve, Belgium
{archambeau, verleysen}@dice.ucl.ac.be

Abstract. In many practical applications, the data is organized along a manifold of lower dimension than the dimension of the embedding space. This additional information can be used when learning the model parameters of Gaussian mixtures. Based on a mismatch measure between the Euclidian and the geodesic distance, manifold constrained responsibilities are introduced. Experiments in density estimation show that manifold Gaussian mixtures outperform ordinary Gaussian mixtures.

1 Introduction

Probability density estimation is a fundamental concept in unsupervised learning and knowledge discovery. In general, density estimation is performed regardless of the intrinsic geometric structure of the data. However, they are concentrated on lower dimensional manifolds embedded in the higher dimensional input space in many data mining applications. As a result, the true density mass in the vicinity of a data point is oriented along the manifold, rather than along all the directions in the input space. Estimating the unknown density by conventional techniques such as the Parzen windows [1] is suboptimal, as it leads to giving too much probability to irrelevant directions of space (i.e. perpendicular to the local manifold orientation) and too little along the manifold. In [2] manifold Parzen windows are introduced to improve nonparametric density estimation in this situation. In this paper, a related approach for mixture models is proposed.

In practice, finite mixtures [3], and in particular Gaussian mixtures, can also be used for nonparametric-like density estimation [4]. That is, provided the number of components can be varied arbitrarily and provided the numerical difficulties encountered when learning the parameters by the expectation-maximization (EM) algorithm [5] can be avoided, they are suitable to estimate any unknown density. The aim of this work is to show how to incorporate the prior knowledge that the data are located on a lower dimensional manifold during the learning process by EM. This is achieved by acting on the responsibilities only. Based on the discrepancy between the Euclidian and the geodesic distance, a manifold constrained E-step is constructed resulting in better generalization capabilities.

* C.A. is supported by the European Commission project IST-2000-25145.
** M.V. is a Senior Research Associate of Belgian National Fund for Scientific Research.

J. Cabestany, A. Prieto, and D.F. Sandoval (Eds.): IWANN 2005, LNCS 3512, pp. 820–828, 2005.

Section 2 presents how to recover the data manifold and how to approximate the geodesic distance by the graph distance. In Section 3, the learning procedure of finite Gaussian mixtures (FGM) by EM is recalled. Section 4 introduces manifold constrained Gaussian mixtures (MFGM) and discusses the resulting E-step. Finally, in Section 5, the approach is validated experimentally and compared to Parzen windows using Gaussian isotropic kernels and ordinary FGM.

2 Constructing the Data Manifold

The basic principle of nonlinear data projection techniques, such as CDA [6] and ISOMAP [7] is to find the lower dimensional data manifold (if any) embedded in the input space and unfold it. An essential building block for constructing the manifold is the geodesic distance. This metric is measured along the manifold and not through the embedding space, akin the Euclidean distance. As a result, the geodesic distance less depends on the curvature of the manifold, thus taking the intrinsic geometrical structure of the data into account.

2.1 Geodesic Distances

Consider two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ on the multidimensional manifold $\mathcal{M}$ of lower dimensionality as the embedding space. Manifold $\mathcal{M}$ is parameterized as follows:

$$\mathbf{m} : \mathbb{R}^p \rightarrow \mathcal{M} \subset \mathbb{R}^d : \mathbf{t} \mapsto \mathbf{x} = \mathbf{m}(\mathbf{t}) \ ,$$

where d is the dimension of the embedding space and p ($\leq d$) is the dimension of $\mathcal{M}$. Different paths may go from point $\mathbf{x}_i$ to point $\mathbf{x}_j$. Each of them is described by a one-dimensional submanifold $\mathcal{P}_{i,j}$ of $\mathcal{M}$ with parametric equations:

$$\mathbf{p} : \mathbb{R} \rightarrow \mathcal{P}_{i,j} \subset \mathbb{R}^p : z \mapsto \mathbf{t} = \mathbf{p}(z) \ .$$

The geodesic distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is then defined as the minimal arc length connecting both data samples:

$$l(\mathbf{x}_i, \mathbf{x}_j) = \min_{\mathbf{p}(z)} \int_{z_i}^{z_j} ||\mathbf{J}_z \mathbf{m}(\mathbf{p}(z))|| dz \ ,$$

where $\mathbf{J}_z(\cdot)$ denotes the Jacobian with respect to z. In practice, such a minimization is untractable, since it is a functional minimization.

2.2 Graph Distances

Even though geodesic distances cannot be computed in practice, they can easily be approximated by graph distances [8]. The problem of minimizing the arc length between two data samples lying on $\mathcal{M}$ reduces to the problem of minimizing the length of path (i.e. broken line) between these samples, while passing through a number of other data points of $\mathcal{M}$. In order to follow the manifold, only the smallest jumps between successive samples are permitted. This can be

achieved by using either the K-rule, or the ϵ-rule. The former allows jumping to the K nearest neighbors. The latter allows jumping to samples lying inside a ball of radius ϵ centered on them. In the remaining of the paper, we only consider the K-rule as the choice for ϵ is more difficult in practice than for K.

The data and the set of allowed jumps constitutes a weighted graph, the vertices being the data, the edges the allowed jumps and the edge labels the Euclidean distance between the corresponding vertices. In order to be a distance, the path length (i.e. the sum of successive jumps) must satisfy the properties of non-negativity, symmetry and triangular inequality. The first and the third are satisfied by construction. Symmetry is ensured when the graph is undirected. For the K-rule, this is gained by adding edges as follows: if $\mathbf{x}_j$ belongs to the K nearest neighbors of $\mathbf{x}_i$, but $\mathbf{x}_i$ is not a neighbor of $\mathbf{x}_j$ then the corresponding edge is added. Remark also that extra edges are added to the graph in order to avoid disconnected parts. For this purpose a minimum spanning tree [9] is used.

The only remaining problem for constructing the distance matrix of the weighted undirected graph is to compute the shortest path between all data samples. This is done by repeatedly applying Dijkstra's algorithm [10], which computes the shortest path between a source vertex and all other vertices in a weighted graph provided the labels are non-negative (which is here the case).

3 Finite Gaussian Mixtures

A finite Gaussian mixture (FGM) [3] is a linear combination of M Gaussian distributions:

$$\hat{p}(\mathbf{x}) = \sum_{m=1}^{M} \pi_m \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m) \ , \tag{1}$$

The mixing proportions $\{\pi_m\}_{m=1}^M$ are non-negative and must sum to one. The multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and precision or inverse covariance matrix $\boldsymbol{\Lambda}$ is defined as:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}) = (2\pi)^{-\frac{d}{2}} |\boldsymbol{\Lambda}|^{\frac{1}{2}} \exp\left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Lambda} (\mathbf{x} - \boldsymbol{\mu}) \right\} \ , \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^d$ and $|\boldsymbol{\Lambda}|$ is the determinant of $\boldsymbol{\Lambda}$.

Estimating the true density $p(\mathbf{x})$ by the approximate density $\hat{p}(\mathbf{x})$ then consists in computing the parameters $\{\boldsymbol{\mu}_m\}_{m=1}^M$, $\{\boldsymbol{\Lambda}_m\}_{m=1}^M$ and $\{\pi_m\}_{m=1}^M$ based on the observed data $\{\mathbf{x}_n\}_{n=1}^N$. By applying the EM algorithm [5] their maximum likelihood estimates can be computed in an elegant way.

Given a particular density model and assuming the data samples are i.i.d., the joint distribution of the observed data or data likelihood is:

$$\mathcal{L} = \hat{p}(\mathbf{x}_1, ..., \mathbf{x}_N | \pi_1, ..., \pi_M, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_M, \boldsymbol{\Lambda}_1, ..., \boldsymbol{\Lambda}_M) = \prod_{n=1}^{N} \hat{p}(\mathbf{x}_n) \ .$$

Unfortunately for FGM, maximizing $\mathcal{L}$ (or equivalently its log) subject to the constraint on the mixture proportions is untractable, unless one defines a component dependent auxiliary variable associated to each data sample:

$$\rho_m(\mathbf{x}_n) = \frac{\pi_m \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)}{\sum_{m=1}^{M} \pi_m \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)} \quad . \tag{3}$$

Keeping the auxiliary variables fixed, the Lagrangian $\log \mathcal{L} + \lambda(\sum_{m=1}^{M} \pi_m - 1)$, λ being the Lagrange multiplier, can be maximized by setting its derivatives with respect to the model parameters to zero. Rearranging leads to the following estimation formulas for the component means, precisions and weights:

$$\boldsymbol{\mu}_m = \frac{\sum_{n=1}^{N} \rho_m(\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} \rho_m(\mathbf{x}_n)} \quad , \tag{4}$$

$$\boldsymbol{\Lambda}_m = \left\{ \frac{\sum_{n=1}^{N} \rho_m(\mathbf{x}_n) \left(\mathbf{x}_n - \boldsymbol{\mu}_m\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_m\right)^{\mathrm{T}}}{\sum_{n=1}^{N} \rho_m(\mathbf{x}_n)} \right\}^{-1} \quad , \tag{5}$$

$$\pi_m = \frac{1}{N} \sum_{n=1}^{N} \rho_m(\mathbf{x}_n) \quad . \tag{6}$$

Observe that (4) and (5) are nothing else than weighted averages based on the auxiliary variables $\rho_m(\mathbf{x}_n)$.

EM [5, 3] operates iteratively in two stages. In the E-step, the auxiliary variables (3) are computed, while the current model parameters are kept fixed. Subsequently, during the M-step the model parameters are updated according to (4-6) using the auxiliary variables computed in the E-step. At each iteration step a monotonic increase of the likelihood function is guaranteed [11].

Interpretation of the E-Step. Each mixture proportion π_m is the prior probability of having the m^{th} component of the mixture. Recalling Bayes' rule, it can easily be seen from expression (3) that each auxiliary variable $\rho_m(\mathbf{x}_n)$ is the posterior probability that data sample $\mathbf{x}_n$ was generated by the mixture component m, provided density model (1). In other words, it corresponds to the probability of having component m if data sample $\mathbf{x}_n$ is observed:

$$\hat{P}(m|\mathbf{x}_n) = \frac{P(m)\hat{p}(\mathbf{x}_n|m)}{\hat{p}(\mathbf{x}_n)} = \frac{\pi_m \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)}{\sum_{m=1}^{M} \pi_m \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)} = \rho_m(\mathbf{x}_n) \quad .$$

The auxiliary variables are therefore often called responsibilities.

Latent Variable Viewpoint of the E-Step. More formally, finite mixture models can be viewed as latent variable models. The component label associated to each data sample is unobserved, that is we do not know by which component a data sample was generated. Consider the set of binary latent vectors $\{\mathbf{z}_n\}_{n=1}^{N}$, with latent variables $z_{nm} \in \{0, 1\}$ indicating which component has

generated $\mathbf{x}_n$ ($z_{nm} = 1$ if $\mathbf{x}_n$ was generated by component m and 0 otherwise, and $\sum_{m=1}^{M} z_{nm} = 1$). The prior distribution of the latent variables and the conditional distribution of observed data are then respectively:

$$\hat{P}(\mathbf{z}_n) = \prod_{m=1}^{M} {\pi_m}^{z_{nm}} \ , \qquad \hat{p}(\mathbf{x}_n|\mathbf{z}_n) = \prod_{m=1}^{M} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)^{z_{nm}} \ .$$

Marginalizing over the latent variables results indeed in (1). Given this latent variable model, it can be shown that EM maximizes iteratively the expected complete data log-likelihood with respect to the posterior distribution of the latent variables (subject to the constraint on the mixture proportions):

$$\mathrm{E}_{z|\mathbf{x}} \left[\log \mathcal{L}\right] = \sum_{n=1}^{N} \sum_{m=1}^{M} \underbrace{\mathrm{E}_{z|\mathbf{x}} \left[z_{nm}\right]}_{\rho_m(\mathbf{x}_n)} \left\{\log \pi_m + \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)\right\} \ ,$$

where $\mathrm{E}_{z|\mathbf{x}} [\cdot]$ is the expectation with respect to $\hat{P}(z_{nm}|\mathbf{x}_n)$. In other words, EM uses the expected value of the latent variables as indicator of the component that generated the data samples. This expected value is equal to the responsibility.

4 Manifold Finite Gaussian Mixtures

Assume the data is lying on a manifold of lower dimension than the dimension of the input space. It would be appealing to take this additional information into account when learning the model parameters. Below, we explain how to achieve this by adjusting the responsibilities according to some prior belief on the discrepancy between the Euclidian and the geodesic distance.

4.1 Manifold Constrained E-Step

Let us respectively denote the Euclidian and graph distance between sample $\mathbf{x}_n$ and component mean $\boldsymbol{\mu}_m$ by $\delta^e(\mathbf{x}_n, \boldsymbol{\mu}_m)$ and $\delta^g(\mathbf{x}_n, \boldsymbol{\mu}_m)$. The graph distance $\delta^g(\mathbf{x}_n, \boldsymbol{\mu}_m)$ approximates the corresponding geodesic distance $l(\mathbf{x}_n, \boldsymbol{\mu}_m)$.

Consider the exponential distribution with location parameter γ and scale parameter β:

$$\mathcal{E}(y|\gamma, \beta) = \frac{1}{\beta} \exp\left\{-\frac{y-\gamma}{\beta}\right\} \ . \tag{7}$$

Setting γ to $\delta^e(\mathbf{x}_n, \boldsymbol{\mu}_m)^2$ and y to $\delta^g(\mathbf{x}_n, \boldsymbol{\mu}_m)^2$ provides an appropriate measure of the mismatch between both distances, since $\delta^e(\mathbf{x}_n, \boldsymbol{\mu}_m) \leq \delta^g(\mathbf{x}_n, \boldsymbol{\mu}_m)$. The adjusted responsibilities can be defined as follows:

$${\rho_m}'(\mathbf{x}_n) = \frac{P(m)\hat{p}'(\mathbf{x}_n|m)}{\hat{p}'(\mathbf{x}_n)} = \frac{\pi_m \mathcal{NE}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)}{\sum_{m=1}^{M} \pi_m \mathcal{NE}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)} \ , \tag{8}$$

where $\mathcal{NE}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)$ is a Gaussian-Exponential distribution of the following particular form:

$$\mathcal{NE}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m) = \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)\mathcal{E}(\delta^g(\mathbf{x}_n, \boldsymbol{\mu}_m)^2|\delta^e(\mathbf{x}_n, \boldsymbol{\mu}_m)^2, 1) \ . \tag{9}$$

Choosing β equal to 1 leaves the responsibility unchanged if both distances are identical. However, when the discrepancy between the distances increases the conditional distribution $\hat{p}'(\mathbf{x}_n|m)$ decreases. This means that it is less likely that data sample $\mathbf{x}_n$ was generated by component m because the corresponding geodesic distance is large compared to the Euclidian distance. This results in a weaker responsibility. As a consequence, data samples lying far away from the component means on the manifold will contribute less to the update of the corresponding component means and precisions during the M-step.

Remark also that adapting the responsibilities in this way is consistent with the latent variable viewpoint. It can be shown that in this case, manifold constrained EM maximizes iteratively the expected complete data log-likelihood with respect to the resulting adjusted posterior $\hat{P}'(z_{nm}|\mathbf{x}_n)$ instead of $\hat{P}(z_{nm}|\mathbf{x}_n)$:

$$\mathrm{E}_{z|\mathbf{x}}\left[\log \mathcal{L}\right] = \sum_{n=1}^{N}\sum_{m=1}^{M} \underbrace{\mathrm{E}_{z|\mathbf{x}}\left[z_{nm}\right]}_{\rho_m{}'(\mathbf{x}_n)} \left\{\log \pi_m + \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)\right\} \quad .$$

In this equation $\mathrm{E}_{z|\mathbf{x}}[\cdot]$ is the expectation with respect to the posterior $\hat{P}'(z_{nm}|\mathbf{x}_n)$, which is adjusted according to the mismatch between both distances.

4.2 Learning Manifold Gaussian Mixtures

The learning procedure for manifold constrained finite Gaussian mixtures (MFGM) can be summarized as follows:

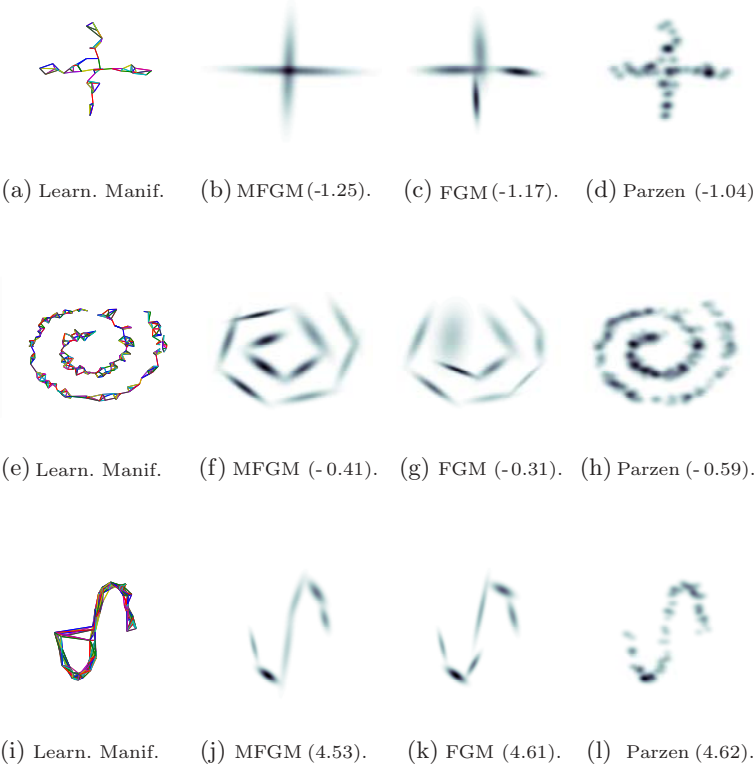
1. Construct the learning manifold by the K-rule and compute the associated distance matrix $\delta^g(\mathbf{x}_i, \mathbf{x}_j)$ by Dijkstra's shortest path algorithm.
2. Repeat until convergence:

   Update the distance matrix of the component means. Find for each $\boldsymbol{\mu}_m$ the K nearest training samples $\{\mathbf{x}_k\}_{k=1}^{K}$ and compute its graph distances to all training data by $\delta^g(\mathbf{x}_n, \boldsymbol{\mu}_m) = \min_k\{\delta^g(\mathbf{x}_n, \mathbf{x}_k) + \delta^e(\mathbf{x}_k, \boldsymbol{\mu}_m)\}$.
   E-step. Compute the manifold constrained responsibilities by (8).
   M-step. Update the model parameters by (4-6).
   End.

Remark that the increase of the computational cost at each iteration step is limited with respect to conventional FGM. Indeed, updating the distance matrix of the component means does not require to recompute the data manifold, nor to re-apply Dijkstra's algorithm. The additional computational effort is due to the construction of the learning manifold and the computation of its distance matrix; both are performed only once (in step 1).

5 Experimental Results

In this section, the quality of MFGM density estimators are assessed on three 2D artificial data sets. MFGM is compared to ordinary FGM and Parzen windows



(a) Learn. Manif. (b) MFGM (-1.25). (c) FGM (-1.17). (d) Parzen (-1.04)

(e) Learn. Manif. (f) MFGM (-0.41). (g) FGM (-0.31). (h) Parzen (-0.59).

(i) Learn. Manif. (j) MFGM (4.53). (k) FGM (4.61). (l) Parzen (4.62).

Fig. 1. Density estimators of a Cross, a Spiral and a S-shape. Each column shows successively the learning manifolds, the estimates of MFGM, ordinary FGM and Parzen windows. For each model, the ANLL of the test set is between parentheses

using Gaussian kernels [1]. The performance measure that we use is the average negative log-likelihood of the test set $\{\mathbf{x}_q\}_{q=1}^{N_t}$: ANLL $= -\frac{1}{N_t}\sum_{q=1}^{N_t}\log\hat{p}(\mathbf{x}_q)$.

The first distribution is a *Cross*. The data samples are generated from a uniform $\mathcal{U}(-0.5,+0.5)$ in horizontal or vertical direction with probability $\frac{1}{2}$. Gaussian noise with zero mean and standard deviation $\sigma_n = 0.03$ is added in the transversal direction. The training set and the validation set contain both 100 samples, and the test set 500 samples. For comparison purposes M is fixed a priori to 4 for both mixture models. The density estimators using the optimal kernel width for Parzen windows ($\sigma_{opt} = 0.03$) and the optimal number of neighbors for MFGM ($K_{opt} = 3$), as well as the ANLL are shown in Figure 1.

The second data set is located along a noisy *Spiral*. A training set of 300 points, a validation set of 300 points and a test of 1000 points were generated from the following distribution: $\mathbf{x} = [0.04t\sin(t) + e_1, -0.04t\cos(t) + e_2]$, where $t \sim \mathcal{U}(3,15)$ and $e_1, e_2 \sim \mathcal{N}(0, 0.025^{-2})$. The number of components in the

mixtures is fixed to 10, the optimal kernel width for Parzen is 0.025 and the optimal number of neighbors for constructing the learning manifold is 4. The results are shown in Figure 1.

The third distribution has a *S-shape*. A training set, validation set a and test set of respectively 100, 100 and 1000 points are generated from one of the following distributions with probability $\frac{1}{2}$: $\mathbf{x} = [3\cos(t) - 3 + e_1, -10\sin(t) + e_2]$ or $\mathbf{x} = [3\cos(t) + 3 + e_1, 10\sin(t) + e_2]$, with $t \sim \mathcal{U}(0, \pi)$ and $e_1, e_2 \sim \mathcal{N}(0, 0.5^{-2})$. The results for $M = 6$, $\sigma_{\text{opt}} = 0.5$ and $K_{\text{opt}} = 10$ are shown in Figure 1.

Discussion. Visually MFGM gives the best results for the three experiments, the discretization step being chosen sufficiently small to avoid visual artifacts. On the one hand, MFGM provides smoother estimates than Parzen windows. On the other hand, the geometric arrangement of the data is better respected with MFGM than with conventional FGM. In the case of the spiral, FGM completely fails to provide a good estimate as one component mixes two branches. Numerically, MFGM generalizes better than FGM in the three examples, as we observe a lower ANLL on the test set (see Fig. 1). Note also that the MFGM is not sensitive to few unhappy edges in the learning manifold, e.g. the S-shape.

6 Conclusion

In this paper, manifold finite Gaussian mixtures (MFGM) were introduced. It was shown that in situations where the data are located along a lower dimensional manifold, MFGM outperforms ordinary FGM. As with FGM, the parameters of MFGM are learnt by EM, except that the E-step is further constrained according to the mismatch between the Euclidean and the geodesic distance. As a result, training samples lying close to a component mean in Euclidean space, but far away on the manifold, will less contribute to the computation of the corresponding mean and covariance matrix in the M-step. In the near future, we plan to extend the approach to other mixtures models, e.g. Student-t mixtures. We also plan to study the effect of fine tuning hyperparameter β, which regulates how the mismatch between both distances penalizes the responsibilities.

References

1. E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Stat.*, 33:1065–1076., 1962.
2. P. Vincent and Y. Bengio. Manifold Parzen windows. In S. Thrun S. Becker and K. Obermayer, editors, *NIPS 15*, pages 825–832. MIT Press, 2003.
3. G. J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, New York, NY., 2000.
4. C. Archambeau and M. Verleysen. From semiparametric to nonparametric density estimation and the regularized Mahalanobis distance. *Submitted.*, 2005.
5. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. Roy. Stat. Soc., B*, 39:1–38., 1977.

6. J. A. Lee, A. Lendasse, and M. Verleysen. Nonlinear projection with curvilinear distances: Isomap versus Curvilinear Distance Analysis. *Neurocomputing*, 57:49–76., 2003.
7. J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323., 2000.
8. M. Bernstein, V. de Silva, J. Langford, and J. Tenenbaum. Graph approximations to geodesics on embedded manifolds. Techn. report Stanford University, CA., 2000.
9. D. B. West. *Introduction to Graph Theory.* Prentice Hall, Upper Saddle River, NJ., 1996.
10. E. W. Dijkstra. A note on two problems in connection with graphs. *Num. Math.*, 1:269–271., 1959.
11. L. Xu and M. I. Jordan. On convergence properties of the EM algorithm for Gaussian mixtures. *Neural Comput.*, 8:129–151., 1996.