Density Estimation with Mercer Kernels

William G. Macready
Research Institute for Advanced Computer Science
NASA Ames Research Center
MailStop 269-4
Moffett Field, CA 94035
wgm@email.arc.nasa.gov

Abstract

We present a new method for density estimation based on Mercer kernels. The density estimate can be understood as the density induced on a data manifold by a mixture of Gaussians fit in a feature space. As is usual, the feature space and data manifold are defined with any suitable positive-definite kernel function. We modify the standard EM algorithm for mixtures of Gaussians to infer the parameters of the density. One benefit of the approach is it's conceptual simplicity, and uniform applicability over many different types of data. Preliminary results are presented for a number of simple problems.

1 Introduction

Kernel methods have proven themselves to be an efficient and effective method for a wide class of machine learning problems. Kernel methods work by mapping data in some space X non-linearly into some feature space F, and applying relatively simple learning methods in the feature space. Historically, most kernel methods have been applied to supervised learning tasks (e.g. support vector machines [1], Gaussian processes [2]), but recent work has expanded their scope into unsupervised problems as well (e.g. kernel PCA [3], support estimation [4], etc).

The prototypical unsupervised learning task is density estimation where we wish to infer a probability density $p_X(\mathbf{x}|D_X)$ that accounts for a data set $D_X = \{\mathbf{x}_i\}_{i=1}^N$ (with $\mathbf{x}_i \in X$)[1]. To date there has been relatively little use of kernel methods for density estimation. Recent exceptions include the use of support vector methods to estimate cumulative distribution functions [5]. Kernel ideas have also inspired improved variations of traditional Parzen window density estimation algorithms [6]. We describe a new approach to density estimation which combines the flexibility and modularity of kernel methods with the simplicity of EM for Gaussian mixtures in order to infer probability densities over any data space, even those having data elements with mixed type (e.g. discrete and continuous).

In kernel methods we assume a mapping $\Phi : X \mapsto F$ taking each datum.to a point in a d_F dimensional Euclidean space F. We indicate the mapped data by $D_F = \{\boldsymbol{\phi}_i\}_{i=1}^N$ where

[1] Scalar values are indicated by variables in regular font, while vectors and matrices are in bold font and indicated by lower- and upper-case letters respectively.

$\phi_i = \Phi(\mathbf{x}_i)$. If X has dimension d_X then Φ maps X into a d_X-dimensional manifold embedded in F (assuming $d_F > d_X$); we call this the data manifold. If the inference algorithm used in the feature space uses only inner products, then the only knowledge required of Φ is contained in the kernel function $K(\mathbf{x}, \mathbf{x}') \equiv \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$.[2] An important advantage of kernel methods is that even though the inference algorithm at work in F operates on vectors in $\mathbb{R}^{d_F}$, the method applies to any type of data X so long as we can identify a Φ mapping the data to F. By now there exist kernels for mapping many types of data, e.g. graphs, trees, symbol sequences, etc., and thus the method proposed here may be used to infer probability densities over all these types of data [7].

If the mapping Φ is suited to the learning task (i.e. the features defined by Φ are relevant), then the inference algorithm at work in F can be very simple. For the density estimation task considered here, we fit a mixture of Gaussian distributions to perform density estimation in the feature space. This choice offers the benefits of modelling flexibility (with enough Gaussians we can approximate any density), and an efficient EM algorithm for determining the parameters of the Gaussians. However, as we will show, even a single Gaussian $p_F(\boldsymbol{\phi}) \sim G(\boldsymbol{\phi}|\boldsymbol{\mu}, \Sigma)$ with $\boldsymbol{\mu}$ and Σ estimated from D_F is often sufficient to model complex structure in p_X. The density p_X is obtained from p_F by simply setting p_X to the the density induced by p_F on the data manifold. This choice does mean the p_X will not be normalized, but we can sample efficiently from p_X (even if it is highly multi-modal)[3], and thus estimate the normalization by Monte Carlo when it is needed.

The paper is organized as follows. In section 2 we derive an EM algorithm to fit mixture of Gaussian densities in feature space by expressing the means and covariances of the Gaussians as linear combinations of $\boldsymbol{\phi}_i$. Optimization of an objective expressible in terms of kernel evaluations gives an update rule to identify the best linear combination. Section 3 then considers how the Gaussian density in F is mapped to a density in X, and section 4 demonstrates some results on simple problems. We conclude in section 4 with a discussion of work in progress and a few open problems.

2 Gaussian Mixture Density Estimation in Feature Space

With M mixture components, the density model in feature space has the form $p_F(\boldsymbol{\phi}|\boldsymbol{\theta}) = \sum_{m=1}^{M} \rho_m G(\boldsymbol{\phi}|\boldsymbol{\theta}_m)$ where $G(\boldsymbol{\phi}|\boldsymbol{\theta}_m) = |2\pi\Sigma_m|^{-1/2} \exp\bigl(-(\boldsymbol{\phi} - \boldsymbol{\mu}_m)^{\mathsf{T}} \Sigma_m^{-1} (\boldsymbol{\phi} - \boldsymbol{\mu}_m)/2\bigr)$. The parameters of the mth mixture are $\boldsymbol{\theta}_m \equiv (\rho_m, \boldsymbol{\mu}_m, \Sigma_m)$, and we group all parameters into the vector $\boldsymbol{\theta} \equiv (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_M)$. The mixture probabilities must sum to 1, i.e. $\sum_{m=1}^{M} \rho_m = 1$. The EM algorithm is a convenient method to determine the parameters $\boldsymbol{\theta}$ of this mixture model. It is an iterative method in which an existing guess for the parameters (call this $\boldsymbol{\theta}^g$) is updated by maximizing the average log posterior of the data D_F. The averaging is done over N hidden variables, z_i, which indicate which mixture was responsible for each observation. If $p(\mathbf{z}|D_F, \boldsymbol{\theta}^g)$ is the the current estimate for the probability of the hidden variables given a guess for the mixture parameters, then define $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^g) \equiv E\bigl(\ln p(D_F, \mathbf{z}|\boldsymbol{\theta}) + \ln p(\boldsymbol{\theta})\bigr)$ where the expectation is performed with respect to $p(\mathbf{z}|D_F, \boldsymbol{\theta}^g)$. For Gaussian mixtures this is calculated as [8]

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^g) = \sum_{m=1}^{M} \left\{ \ln p(\boldsymbol{\theta}_m) + \sum_{i=1}^{N} \ln(\rho_m G(\boldsymbol{\phi}_i|\boldsymbol{\theta}_m) p(m|\boldsymbol{\phi}_i, \boldsymbol{\theta}^g) \right\}$$

[2] $\langle \boldsymbol{\phi}, \boldsymbol{\phi}' \rangle$ is the usual inner product $\sum_{\alpha=1}^{d_F} \phi(\alpha)\phi'(\alpha)$ where $\phi(\alpha)$ is the αth component of $\boldsymbol{\phi}$. If F is infinite dimensional the sum is replaced by an integral.

[3] Work in progress.

where using Bayes rule

$$p(m|\boldsymbol{\phi}_i, \boldsymbol{\theta}^g) = \frac{\rho_m^g G(\boldsymbol{\phi}_i|\boldsymbol{\theta}_m^g)}{\sum_{m'} \rho_{m'}^g G(\boldsymbol{\phi}_i|\boldsymbol{\theta}_{m'}^g)} \qquad (1)$$

is the probability that $\boldsymbol{\phi}_i$ was generated by the mth mixture. Given the existing guess $\boldsymbol{\theta}^g$ a better guess is obtained by maximizing $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^g)$ with respect to $\boldsymbol{\theta}$. In this formulation we have allowed for prior probabilities, $p(\boldsymbol{\theta}_m)$, on the mixture parameters. This inclusion is essential in order to guarantee that we obtain positive-definite covariance estimates. For many kernels, $d_F > N$ so that naive estimation of Σ_m would result in singular covariances. We will not have need of priors over the means and mixture weights, and we employ an inverse Wishart distribution for the prior over each $\boldsymbol{\Sigma}_m$ [9]. The inverse Wishart distribution is given by $p(\Sigma_m|\alpha, \beta, \boldsymbol{J}) \propto |\Sigma_m^{-1}|^{\beta/2} \exp\bigl(-\alpha \operatorname{tr}(\Sigma_m^{-1}\mathbf{J})/2\bigr)$. The role of Wishart parameters can be seen by maximizing the inverse Wishart distribution. The mode occurs at $\Sigma_m = \alpha\mathbf{J}/\beta$. In what follows we will take $\mathbf{J} = \mathbf{I}$, the identity in feature space. Up to irrelevant constants, $\log p(\Sigma_m) = (\beta \ln|\Sigma_m^{-1}| - \alpha \operatorname{tr} \Sigma_m^{-1})/2$. Defining

$$n_m^g \equiv \sum_{i=1}^{N} p(m|\boldsymbol{\phi}_i, \boldsymbol{\theta}_m^g) \quad \text{and} \quad \mathbf{S}_m \equiv \sum_{i=1}^{N} (\boldsymbol{\phi}_i - \boldsymbol{\mu}_m)(\boldsymbol{\phi}_i - \boldsymbol{\mu}_m)^\top p(m|\boldsymbol{\phi}_i, \boldsymbol{\theta}^g) \qquad (2)$$

a standard calculation yields

$$Q(\theta|\theta^g) = \sum_{m=1}^{M} \left\{ n_m^g \ln \rho_m + \frac{n_m^g + \beta}{2} \ln|\boldsymbol{\Sigma}_m^{-1}| - \frac{1}{2} \operatorname{tr}(\boldsymbol{\Sigma}_m^{-1}\mathbf{S}_m + \alpha\boldsymbol{\Sigma}_m^{-1}) \right\}.$$

Since the EM algorithm only has access to inner products in F we must express Q in terms of the kernel K. To this end we write the mean and covariance as: $\boldsymbol{\mu}_m = \mathbf{V}\mathbf{a}_m$ and $\Sigma_m = \epsilon_m\mathbf{I} + \mathbf{V}\mathbf{B}_m\mathbf{V}^\top$ where $\mathbf{V} = [\boldsymbol{\phi}_1 \ \cdots \ \boldsymbol{\phi}_N]$. The parameters ϵ_m, $\mathbf{a}_m$ and $\mathbf{B}_m$ that we need to determine are respectively a scalar, an N-vector and a positive definite $N \times N$ matrix. With a slight abuse of notation we set $\boldsymbol{\theta}_m = (\epsilon_m, \mathbf{a}_m, \Sigma_m)$ which we will determine by maximizing Q. We include a multiple of the identity to ensure that Σ_m is positive-definite. It is easily verified that the inverse of the covariance is given by $\Sigma_m = \tilde{\epsilon}_m\mathbf{I} + \mathbf{V}\tilde{\mathbf{B}}_m\mathbf{V}^\top$ if

$$\tilde{\epsilon}_m = 1/\epsilon_m \quad \text{and} \quad \tilde{\mathbf{B}}_m = -\tilde{\epsilon}_m\mathbf{B}_m^{1/2}(\epsilon_m\mathbf{I}_N + \mathbf{B}_m^{1/2}\mathbf{K}\mathbf{B}_m^{1/2})^{-1}\mathbf{B}_m^{1/2}$$

where $\mathbf{I}_N$ is the $N \times N$ identity matrix, $\mathbf{K}$ is the symmetric positive definite $N \times N$ Gram matrix given by $\mathbf{K} = \mathbf{V}^\top\mathbf{V} = [K_{i,j}]$ with $K_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathbf{B}_m^{1/2}$ is the Cholesky decomposition of $\mathbf{B}_m$. For future reference we also note the identities

$$\epsilon_m\tilde{\mathbf{B}}_m + \tilde{\epsilon}_m\mathbf{B}_m + \tilde{\mathbf{B}}_m\mathbf{K}\mathbf{B}_m = \epsilon_m\tilde{\mathbf{B}}_m + \tilde{\epsilon}_m\mathbf{B}_m + \mathbf{B}_m\mathbf{K}\tilde{\mathbf{B}}_m = 0.$$

From these equations we may derive

$$\tilde{\epsilon}_m\mathbf{K}^{-1} + \tilde{\mathbf{B}}_m = (\epsilon_m\mathbf{K} + \mathbf{K}\mathbf{B}_m\mathbf{K})^{-1} \quad \text{and} \quad \epsilon_m\mathbf{K}^{-1} + \mathbf{B}_m = (\tilde{\epsilon}_m\mathbf{K} + \mathbf{K}\tilde{\mathbf{B}}_m\mathbf{K})^{-1} \qquad (3)$$

which generalizes from inverses to pseudoinverses.

With the assumed representations for $\boldsymbol{\mu}_m$ and Σ_m, the argument of the exponential in the mth Gaussian when evaluated at $\boldsymbol{\phi}_x \equiv \Phi(\mathbf{x})$ is $(\boldsymbol{\phi}_x - \mu_m)^\top\Sigma_m^{-1}(\boldsymbol{\phi}_x - \mu_m) = (\boldsymbol{\phi}_x - \boldsymbol{V}\mathbf{a}_m)^\top(\tilde{\epsilon}_m\mathbf{I} + \mathbf{V}\tilde{\mathbf{B}}_m\mathbf{V}^\top)(\boldsymbol{\phi}_x - \mathbf{V}\mathbf{a}_m)$ which is equal to

$$\begin{aligned} &= (\mathbf{k}_x - \mathbf{K}\mathbf{a}_m)^\top\tilde{\mathbf{B}}_m(\mathbf{k}_x - \mathbf{K}\mathbf{a}_m) + \tilde{\epsilon}_m(K_{x,x} - 2\mathbf{k}_x^\top\mathbf{a}_m + \mathbf{a}_m^\top\mathbf{K}\mathbf{a}_m) \\ &= \tilde{\epsilon}_m(K_{x,x} - \mathbf{k}_x^\top\mathbf{K}^{-1}\mathbf{k}_x) + (\mathbf{k}_x - \mathbf{K}\mathbf{a}_m)^\top(\tilde{\mathbf{B}}_m + \tilde{\epsilon}_m\mathbf{K}^{-1})(\mathbf{k}_i - \mathbf{K}\mathbf{a}_m) \end{aligned}$$

where we have defined $K_{x,x} \equiv K(\mathbf{x}, \mathbf{x})$, and $\mathbf{k}_x^\top \equiv \boldsymbol{\phi}_x^\top \mathbf{V} = [K(\mathbf{x}, \mathbf{x}_1) \quad \cdots \quad K(\mathbf{x}, \mathbf{x}_N)]$. If $\mathbf{x}$ is the ith data point then $\mathbf{k}_i^\top \mathbf{K}^{-1} \mathbf{k}_i = K_{i,i} = K(x_i, x_i)$ so that using Eq. (3) the argument of the exponential can be written as

$$(\boldsymbol{\phi}_i - \boldsymbol{\mu}_m)^\top \Sigma_m^{-1} (\boldsymbol{\phi}_i - \boldsymbol{\mu}_m) = (\mathbf{k}_i - \mathbf{K}\mathbf{a}_m)^\top (\epsilon_m \mathbf{K} + \mathbf{K}\mathbf{B}_m\mathbf{K})^{-1} (\mathbf{k}_i - \mathbf{K}\mathbf{a}_m). \tag{4}$$

Returning to the expression of Q in terms of inner products, we can show that $|\boldsymbol{\Sigma}_m^{-1}| = \tilde{\epsilon}_m^{d_F - N} |\mathbf{K}| |\tilde{\epsilon}_m \mathbf{K}^{-1} + \tilde{\mathbf{B}}_m|$. Thus $\ln |\boldsymbol{\Sigma}_m^{-1}|$ which contributes to $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^g)$ is

$$\ln |\boldsymbol{\Sigma}_m^{-1}| = (d_F - N) \ln \tilde{\epsilon}_m + \ln |\mathbf{K}| + \ln |\tilde{\epsilon}_m \mathbf{K}^{-1} + \tilde{\mathbf{B}}_m|$$

Further, exploiting Eq. (3) in Eq. (1) we see that

$$p(m|\boldsymbol{\phi}_i, \boldsymbol{\theta}^g) = \frac{\rho_m^g |\epsilon_m \mathbf{K} + \mathbf{K}\mathbf{B}_m\mathbf{K}|^{-1/2} \exp\left(-\frac{1}{2}(\boldsymbol{\phi}_i - \boldsymbol{\mu}_m)^\top \Sigma_m^{-1} (\boldsymbol{\phi}_i - \boldsymbol{\mu}_m)\right)}{\sum_m \rho_m^g |\epsilon_m \mathbf{K} + \mathbf{K}\mathbf{B}_m\mathbf{K}|^{-1/2} \exp\left(-\frac{1}{2}(\boldsymbol{\phi}_i - \boldsymbol{\mu}_m)^\top \Sigma_m^{-1} (\boldsymbol{\phi}_i - \boldsymbol{\mu}_m)\right)} \tag{5}$$

with the arguments of the exponentials given by Eq. (4). Thus, the posterior probabilities $p(m|\boldsymbol{\phi}_i, \boldsymbol{\theta}^g)$ can be evaluated in terms of kernel values so that $\mathbf{S}_m$ defined in Eq. (2) can be determined.

Finally, we consider $\operatorname{tr}(\Sigma_m^{-1} \mathbf{S}_m + \alpha \Sigma_m^{-1})$ which also appears in Q. From the defining equation for $\boldsymbol{\Sigma}_m^{-1}$

$$\operatorname{tr} \boldsymbol{\Sigma}_m^{-1} = \tilde{\epsilon}_m \operatorname{tr} \mathbf{I} + \operatorname{tr}(\mathbf{V}\tilde{\mathbf{B}}_m\mathbf{V}^\top) = \tilde{\epsilon}_m d_F + \operatorname{tr}(\tilde{\mathbf{B}}_m \mathbf{V}^\top \mathbf{V}) = \tilde{\epsilon}_m d_F + \operatorname{tr}(\tilde{\mathbf{B}}_m \mathbf{K}).$$

Similarly, $\operatorname{tr}(\Sigma_m^{-1}\mathbf{S}_m) = \tilde{\epsilon}_m \operatorname{tr} \mathbf{S}_m + \operatorname{tr} \mathbf{V}\tilde{\mathbf{B}}_m\mathbf{V}^\top \mathbf{S}_m = \tilde{\epsilon}_m \operatorname{tr} \mathbf{S}_m + \operatorname{tr} \tilde{\mathbf{B}}_m \mathbf{V}^\top \mathbf{S}_m \mathbf{V}$. Using the definition for $\mathbf{S}_m$, and the fact that $\boldsymbol{\mu}_m = \mathbf{V}\mathbf{a}_m$ we find

$$\begin{aligned} \operatorname{tr} \mathbf{S}_m &= \sum_{i=1}^{N} p(m|\boldsymbol{\phi}_i, \boldsymbol{\theta}^g) \operatorname{tr}(\boldsymbol{\phi}_i - \mathbf{V}\mathbf{a}_m)(\boldsymbol{\phi}_i - \mathbf{V}\mathbf{a}_m)^\top = \sum_{i=1}^{M} p(m|\boldsymbol{\phi}_i, \boldsymbol{\theta}^g)(\boldsymbol{\phi}_i - \mathbf{V}\mathbf{a}_m)^\top (\boldsymbol{\phi}_i - \mathbf{V}\mathbf{a}_m) \\ &= \sum_{i=1}^{N} p(m|\boldsymbol{\phi}_i, \boldsymbol{\theta}^g)(\mathbf{k}_i - \mathbf{K}\mathbf{a}_m)^\top \mathbf{K}^{-1} (\mathbf{k}_i - \mathbf{K}\mathbf{a}_m) = \operatorname{tr}\left(\mathbf{K}^{-1}\mathbf{M}_m\right). \end{aligned}$$

where we have defined the $N \times N$ matrix $\mathbf{M}_m \equiv \sum_i p(m|\boldsymbol{\phi}_i, \boldsymbol{\theta}^g)(\mathbf{k}_i - \mathbf{K}\mathbf{a}_m)(\mathbf{k}_i - \mathbf{K}\mathbf{a}_m)^\top$. The final term, $\operatorname{tr} \tilde{\mathbf{B}}_m \mathbf{V}^\top \mathbf{S}_m \mathbf{V}$, is expressed by noting that $\mathbf{V}^\top \mathbf{S}_m \mathbf{V} = \sum_i p(m|\boldsymbol{\phi}_i, \boldsymbol{\theta}^g) \mathbf{V}^\top (\boldsymbol{\phi}_i - \mathbf{V}\mathbf{a}_m)(\boldsymbol{\phi}_i - \mathbf{V}\mathbf{a}_m)^\top \mathbf{V} = \mathbf{M}_m$. Combining these results we find $\operatorname{tr}(\Sigma_m^{-1}\mathbf{S}_m + \alpha \Sigma_m^{-1}) = \operatorname{tr}\left((\tilde{\epsilon}_m \mathbf{K}^{-1} + \tilde{\mathbf{B}}_m)(\mathbf{M}_m + \alpha \mathbf{K})\right) + \alpha \tilde{\epsilon}_m (d_F - N)$ so that $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^g)$ is equal to (up to constants independent of $\tilde{\epsilon}_m$, $\mathbf{a}_m$, $\tilde{\mathbf{B}}_m$)

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^g) = \sum_{m=1}^{M} \Big\{ & n_m^g \ln \rho_m - \frac{\alpha(d_F - N)\tilde{\epsilon}_m}{2} + \frac{n_m^g + \beta}{2} \left(\ln \tilde{\epsilon}_m^{d_F - N} + \ln |\tilde{\epsilon}_m \mathbf{K}^{-1} + \tilde{\mathbf{B}}_m|\right) - \\ & \frac{1}{2} \operatorname{tr}\left((\tilde{\epsilon}_m \mathbf{K}^{-1} + \tilde{\mathbf{B}}_m)(\mathbf{M}_m + \alpha \mathbf{K})\right) \Big\} \end{aligned} \tag{6}$$

Maximizing this with respect to $\boldsymbol{\theta}$ determines the update formulas for $\tilde{\epsilon}_m$, $\mathbf{a}_m$, and $\tilde{\mathbf{B}}_m$.

2.1 Maximization of Q

To update ρ_m we maximize Eq. (6) with respect to ρ_m subject to the constraint that $\sum_m \rho_m = 1$. This yields

$$\rho_m = \frac{n_m^g}{N} = \frac{1}{N} \sum_{i=1}^{N} p(m|\boldsymbol{\phi}_i, \boldsymbol{\theta}_m^g). \tag{7}$$

Similarly setting variations dQ of Q with respect to variations $d\mathbf{M}_m$ equal to zero gives

$$dQ = -\frac{1}{2}\,\mathrm{tr}\big((\tilde{\epsilon}_m\mathbf{K}^{-1} + \tilde{\mathbf{B}}_m)d\mathbf{M}_m\big).$$

Thus the optimal $\mathbf{a}_m$ is determined by $d\mathbf{M}_m = 0$. Using the definition of $\mathbf{M}_m$ we find $\mathbf{a}_m$ is determined from $d\mathbf{M}_m = 2\sum_i p(m|\boldsymbol{\phi}_i, \boldsymbol{\theta}^g)(\mathbf{a}_m^\top\mathbf{K} - \mathbf{k}_i^\top)d\mathbf{a}_m = 0$ which has solution

$$\mathbf{K}\mathbf{a}_m = \sum_i \frac{p(m|\boldsymbol{\phi}_i, \boldsymbol{\theta}_m^g)}{n_m^g}\mathbf{k}_i \quad \text{or} \quad \mathbf{a}_m = \sum_i \frac{p(m|\boldsymbol{\phi}_i, \boldsymbol{\theta}_m^g)}{n_m^g}\mathbf{e}_i. \tag{8}$$

where $\mathbf{e}_i$ is a unit vector in the ith direction.

Next we maximize Q with respect to $\tilde{\mathbf{B}}_m$ to find the parameters for the covariance. Rather than consider variations in Q with respect to variations in $\tilde{\mathbf{B}}_m$ we consider equivalently variations in $\mathbf{T} \equiv \tilde{\epsilon}_m\mathbf{K}^{-1} + \tilde{\mathbf{B}}_m$:

$$dQ = \frac{n_m^g + \beta}{2}d(\ln|\mathbf{T}|) - \frac{1}{2}\,\mathrm{tr}\big((d\mathbf{T})(\mathbf{M}_m + \alpha\mathbf{K})\big).$$

The variation in the log determinant is given by $d(\ln|\mathbf{T}|) = \mathrm{tr}\big(\mathbf{T}^{-1}(d\mathbf{T})\big)$ so that

$$dQ = \frac{1}{2}\,\mathrm{tr}\left(\left((n_m^g + \beta)\mathbf{T}^{-1} - \mathbf{M}_m - \alpha\mathbf{K}\right)(d\mathbf{T})\right).$$

Thus $\mathbf{T}^{-1} = (\mathbf{M}_m + \alpha\mathbf{K})/(n_m^g + \beta)$. Recalling that $\mathbf{T} = \tilde{\epsilon}_m\mathbf{K}^{-1} + \tilde{\mathbf{B}}_m$ we find

$$(\tilde{\epsilon}_m\mathbf{K}^{-1} + \tilde{\mathbf{B}}_m)^{-1} = \epsilon_m\mathbf{K} + \mathbf{K}\mathbf{B}_m\mathbf{K} = \frac{1}{n_m^g + \beta}(\mathbf{M}_m + \alpha\mathbf{K})$$

where we have utilized identity Eq. (3). Maximizing Q with respect to ϵ_m yields $\epsilon_m = \alpha/(n_m^g + \beta)$. This last result for ϵ_m gives a simple expression for $\mathbf{B}_m$,

$$\mathbf{B}_m = \sum_{i=1}^{N} \frac{p(m|\phi_i, \theta_m^g)}{n_m^g + \beta}(\mathbf{e}_i - \mathbf{a}_m)(\mathbf{e}_i - \mathbf{a}_m)^\top. \tag{9}$$

The complete EM updates are thus specified with equations Eqs. (7), (8), (9). The EM iterations can be initialized with a K-means algorithm in feature space.

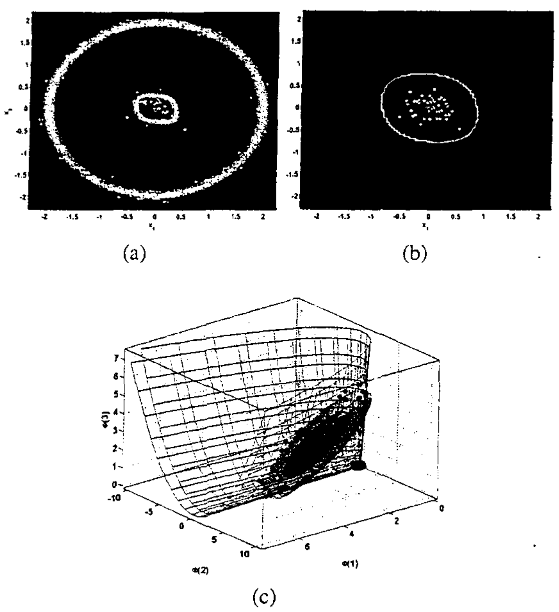
Singular K: In many cases the above equations may not be directly applicable because $\mathbf{K}$ is singular (i.e. some $\boldsymbol{\phi}_i$ are linearly dependent, e.g. when $N > d_F$). Thus $\mathbf{V}$ is effectively a $d_F \times r$ matrix where $r < N$ is the number of linearly independent $\boldsymbol{\phi}_i$ vectors (and the rank of $\mathbf{K}$). This means that rather than using the full inverse of $\mathbf{K}$ we use it's pseudoinverse. Since Eq. (3) also holds for pseudoinverses, we perform all calculations in an r dimensional subspace of F corresponding to the non-singular eigenvectors of $\mathbf{K}$ rather than the full N dimensional subspace.

Shifting the Origin in Feature Space: If the $\{\boldsymbol{\phi}_i\}$ are far from the origin in feature space, the Gram matrix conveys much less useful information. In some applications, shifting the origin in feature space to the center of mass $\mathbf{c} = \sum_{i=1}^N \phi_i/N$ can improve performance [10]. This is accomplished in the present context by expanding the mean and covariance matrix in terms of $\boldsymbol{\phi}_i^c \equiv \phi_i - \mathbf{c}$. Because translations preserve inner products, the argument of the exponentials is unaltered if $\mathbf{c}$ is absorbed into $\mathbf{a}_m$. Thus the EM algorithm can be applied as before with new definitions for $\mathbf{K}^c$ and $\mathbf{k}_x^c$.

3 From Feature Space to Data Space

Having determined the density p_F in feature space we obtain the density in X by evaluating p_F on the data manifold in F.[4] Explicitly, $p_X(\mathbf{x}|D_X) \propto p_F(\Phi(\mathbf{x})|\boldsymbol{\theta})$. As noted earlier

[4] We might have induced a density on the data manifold as $p_X(x) = \int d\boldsymbol{\phi}\,\delta\big(\mathbf{x} - \Phi^{-}(\boldsymbol{\phi})\big)p_F(\boldsymbol{\phi})$ where Φ^{-} is a surjection from F to the data manifold, but this would result in non-analytic forms.



(a) (b)

(c)

Figure 1: (a) Probability density obtained with two Gaussians fit to 100 noisy samples (white dots) centered at the origin and in a circle of radius 2. Blue regions are low probability and high regions are red. (b) Classification boundary obtained from the posterior probability $p(m|\phi_x)$; red indicates class 1 (white samples) while blue indicates class 2 (black samples). The pair of Gaussian densities in feature space. The 2 dimensional data manifold is shown in the wire mesh with black dots representing the data samples. Four isosurfaces of constant probability are shown in varying translucencies of red.

this density is not normalized. In fact, for kernel functions which are local and defined over infinite spaces, i.e. $K(\mathbf{x}, \mathbf{x}') \to 0$ as $\|\mathbf{x} - \mathbf{x}'\| \to \infty$ the density asymptotes to a tiny but finite value and cannot be normalized. In spite of this unpleasant property we have found good results even for Gaussian kernels.

4 Experiments

As a first example we demonstrate the geometry underlying our approach for $X = \mathbb{R}^2$. In order to visualize the results we select the quadratic kernel $K(\mathbf{x}, \mathbf{x}') = [x(1)x'(1) + x(2)x'(2)]^2$ where $x(i)$ is the ith component of $\mathbf{x}$. For this kernel one choice for the mapping to feature space is $\Phi(x) = [\Phi(1)\ \Phi(2)\ \Phi(3)]^\top = [x(1)^2\ \sqrt{2}x(1)x(2)\ x(1)^2]^\top$ for which $d_F = 3$. In Figure 1(a) we plot the density obtained using a 2 component Gaussian mixture fit to a simple illustrative data set of 100 data points (white dots of Figure 1(a)). One Gaussian captures data near the origin while the other captures the halo around the origin, see Figure 1(c). Regularization of the density is controlled by the α and β parameters of the inverse Wishart prior over the covariance. In this example $\alpha = \beta = 1$. Larger values of these parameters result in more spherical covariance matrices and smoother density estimates. In Figure 1(b) we also plot $p(m = 1|\phi_x)$ for varying $\mathbf{x}$. High values (red) indicate points assigned to the cluster at the origin and low values (blue) indicate points assigned to

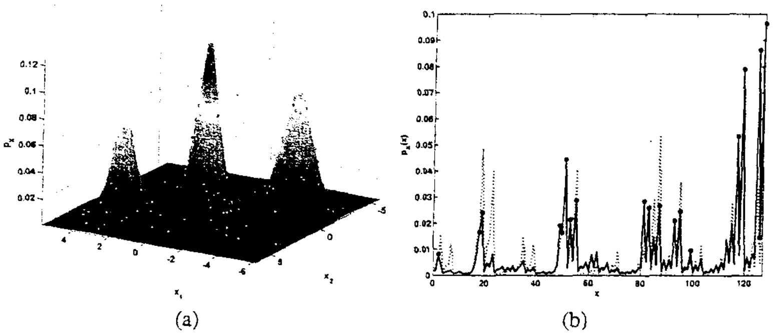


Figure 2: (a) Fit obtained with a single Gaussian in feature space to 200 samples from two Gaussians and a Laplace distribution. (b) Fit obtained to a sample of 30 bit strings drawn from a mixture of 2 equally weighted Bernoulli distributions. The solid black line is the fit with shaded circles showing where data points were located, and the true density is shown as dashed blue. The spiky nature of the plot is due to representing each bit string by its decimal equivalent.

the halo. The data has been colored white or black according to its true class label.

Next, we consider 200 samples from a mixture density in $\mathbb{R}^2$ consisting of two Gaussians (with means $-[2\,2]$ and $[2\,2]$, weights 0.3 and 0.3, and identity covariance), and one Laplacian density (with weight 0.4) at the origin, $p_L(\mathbf{x}) = |\mathbf{C}^{-1}|\exp(\|\mathbf{C}^{-1}\mathbf{x}\|_1)/2^{d_X}$ with covariance $\mathbf{C} = [1\ -0.6; -0.6\ 1]$. A single Gaussian in feature space with $\alpha = 15$ and $\beta = 1$ and the kernel $K(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2))$ results in the estimate shown in 2(a). The estimate captures the sharp peak and fat tails of the Laplacian.

As a final example we apply the method to estimate a density over a discrete space. For easy visualization[5] we chose $X = \mathbb{B}^7$, and used the kernel $K(\mathbf{x}, \mathbf{x}') = \rho^{d(\mathbf{x},\mathbf{x})}$ where $d(\mathbf{x}, \mathbf{x}')$ is the Hamming distance between the bit strings $\mathbf{x}$, and $\mathbf{x}'$, and $-1 \leq \rho \leq 1$ is a hyperparameter [7]. The results are not terribly sensitive to the value of ρ (as long as ρ is positive). In Figure 2(b) we plot the fit obtained with two Gaussians to 30 samples (19 of which are distinct) from a mixture of two equally weighted Bernoulli distributions[6] ρ and β were arbitrarily set to 0.6 and 0.5 respectively, and α was set to 2.6 by leave one out cross-validation. The fit is very good, even where the estimate is high or low, it captures the change in probability as bits are flipped accurately.

5 Discussion

We have developed a conceptually simple density estimation procedure which works by fitting a mixture of Gaussian distributions in feature space, and using the density induced on the data manifold. The EM algorithm can be modified to easily determine the parameters of the Gaussians. Preliminary results on simple test problems are encouraging. The method scales well with the dimensionality of the data space d_X, but poorly with the number of training examples N. It would be useful to adapt methods from other kernel methods to

[5] Higher dimensions d_X only affect the scaling of the algorithm through evaluation of the kernel function.

[6] The probability of bit i being 1 was [0.9501 0.6068 0.8913 0.4565 0.8214 0.6154 0.9218] and [0.2311 0.4860 0.7621 0.0185 0.4447 0.7919 0.7382] for the two mixtures.

choose good subsets of the data to improve the scaling with N [11]. The other important improvement that should be made concerns the determination of hyperparameters. Currently, this is difficult because we do not have access to the normalization of the density which depends on the hyperparameters. For general applicability, a method to overcome this difficulty to automatically identify hyperparameters is desirable.

Though we have not outlined the details here, it is straightforward to modify the algorithm to account for data spaces having mixed types, e.g. discrete and continuous elements. This generalization will be reported elsewhere. The Gaussian description in feature space brings with it significant advantages. Firstly, because Gaussians are simple to sample from, we may be able use this to sample efficiently from $p_X(\mathbf{x})$. We are currently fleshing this idea out. Secondly, classification/regression can be done by fitting a single Gaussian to the joint space $\mathbf{x}$ and $\mathbf{y}$, and determining $p_{F_y|F_x}(\phi_y|\phi_x)$. This induces a (typically non-Gaussian) density $p_{Y|X}(\mathbf{y}|\mathbf{x})$ over the data manifold.

Acknowledgements

We would like to thank E. Bandari, A. Srivastava and D. Wolpert for helpful suggestions.

References

[1] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[2] C. K. I. Williams. Prediction with gaussian processes: From linear regression to linear prediction and beyond. In M. I. Jordan, editor, *Learning and Inference in Graphical Models*. Kluwer, 1998.

[3] B. Schölkopf, A. Smola, and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299 – 1319, 1998.

[4] B. Schölkopf, B. Platt, J. ShaweTaylor, A. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443 – 1471, 2001.

[5] V. Vapnik and S. Mukherjee. Support vector method for multivariate density estimation. In S. Solla, T. Leen, and K.-R. Muller, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 659 – 665. MIT Press, 2000.

[6] M. Girolami and C. He. Probability density estimation from optimally condensed data samples. *IEEE Transaction on Pattern Analysis and Machine Learning*, 2003. In press.

[7] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Machine Learning: Proceedings of the 19th International Conference*, 2002.

[8] J. A. Bilmes. A gentle tutorial of the em algorithm and its applications to parameter estimation for gaussian mixture and hidden markov models. Technical Report TR-97-021, International Computer Science Institute, Berkeley, Ca., 1998.

[9] H. Snoussi and A. Mohammad-Djafari. Penalized maximum likelihood for multivariate gaussian mixture. In R. L. Fry and M. Bierbaum, editors, *Bayesian Inference and Maximum Entropy Methods in Science and Engineering*. American Institute of Physics, 2001.

[10] M. Meila. Data centering in feature space. In C. M. Biship and B. J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, volume 8, pages 331–337, 2003.

[11] M. Seeger, C. K.I. Williams, and N. D. Lawrence. Fast forward selection to speed up sparse gaussian process regression. In C. M. Biship and B. J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, volume 8, 2003.