MAXIMUM ENTROPY APPROXIMATION FOR KERNEL MACHINES

Umut Ozertem, Deniz Erdogmus
CSEE Department, Oregon Health and Science University, Portland, Oregon, USA

ABSTRACT

Kernel machines are widely used in pattern recognition, exploratory data analysis, and statistical signal processing, due to their effectiveness of modeling nonlinear dependencies in the data. The computational burden in evaluating forward functions in testing is the main drawback for kernel machines, especially in high dimensional large training set situations. We present a separable maximum entropy approximation for kernel machines that reduce the computational load for forward function evaluation. The performance of the approximation is demonstrated on kernel-based discriminative nonlinear projections on benchmark datasets.

1. INTRODUCTION

Kernel methods provide a convenient and principled approach to training nonlinear function approximators using convex optimization techniques. They have been motivated theoretically by the existence of a nonlinear eigenfunction mapping that moves the nonlinear problem to a high dimensional Hilbert space, where the problem is solved using linear techniques. Overall, training and testing procedures are carried out using inner products in the high dimensional space, which translate to the original data space as pairwise kernel evaluations. Kernel principal components analysis (KPCA) [1], and kernel linear discriminant analysis (KLDA) [2] are widely known examples of kernel machines, besides the celebrated support vector machines (SVM).

Kernel machines, being nonparametric function approximators, suffer from high computational complexity in forward function evaluation in testing phase. Kernel machines typically require the evaluation of the kernel between the novel test data and all available training data, this is detrimental to fast computation for large training sets. The general form of the kernel machine is:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i K_i(\mathbf{x} - \mathbf{x}_i) \tag{1}$$

where the $\mathbf{x}_i$ are the training samples and α_i are the associated weights. For the purposes of this paper, how the weights are obtained is irrelevant.

Our objective is to exploit the structure of the input feature space to reduce the computation and storage requirements of the kernel methods. The approximation involves two steps: (i) data clustering in the training set to determine compact partitions, (ii) a maximum entropy based separable function approximation for the portion of the function corresponding to each cluster. Clustering will be achieved by a fixed-point mode-seeking algorithm similar to mean shift clustering [3]. Parametric approximation of each mode is then achieved by a product of exponential functions whose parameters are optimized using independent component analysis for decomposition and maximum entropy principle for parameter fitting in each *independent* dimension.

2. FIXED POINT MODE SEEKING

Given the kernel function $K_i(.)$, the data points $\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N\}$, and the corresponding weights $\{\alpha_1,\alpha_2,\ldots,\alpha_N\}$, we need to partition the data into compact clusters that contribute to each mode of the kernel machine given in (1). This can be achieved by initializing a mode seeking algorithm to each training data sample and letting it converge to the mode that it most contributes to. Note that the modes are basically the positive and negative *bumps* in the function $f(\mathbf{x})$. The samples that converge to the same mode are put into the same cluster. Certain approximation methods (such as the Fast Gauss transform [4]) consider clustering the data without considering the weights, thus are ignorant to the importance of each sample when determining the approximation centers. Our approach departs from this common oversight.

A mean shift like fixed point algorithm is utilized to determine the mode corresponding to each training sample. The iterations for each particle initialized to the training samples is simply obtained by equating the gradient of the expression in (1) to zero and manipulating terms to obtain a fixed point update rule. For the most common Gaussian kernel, this update rule becomes:

$$\mathbf{x} \leftarrow \frac{\sum_{i=1}^{N} \alpha_i G_i(\mathbf{x} - \mathbf{x}_i)\, \mathbf{x}_i}{\sum_{i=1}^{N} \alpha_i G_i(\mathbf{x} - \mathbf{x}_i)} \tag{2}$$

In the context of mean shift clustering, this algorithm is known to behave as an EM-type update rule with linear

convergence rate [4]. Although we do not prove the same here, it is inituitively expected that this algorithm exhibit similar convergence speed. The fundamental difference here is that the fixed point update in (2) is utilized to find both concave and convex peaks/modes of the kernel machine $f(\mathbf{x})$.

Figure 1 depicts an illustrative kernel machine on a one-dimensional dataset. This example helps illustrate the multi-mode structure of a typical kernel machine and how the algorithm in (2) can be used to identify positive and negative modes of the function to place the approximation centers. One advantage of the fixed point approach is that the algorithm decides whether ascent or descent direction is to be pursued given the particular initializing training sample; further no parameters need to be fine tuned or randomized. Therefore, the clustering procedure is completely deterministic and automatic.

Once all training samples are iterated through (2) until a clustering solution is achieved, the data is partitioned into C partitions consisting of data points $\{\mathbf{x}_1^c,...,\mathbf{x}_{N_c}^c\}_{c=1}^C$. Consequently, the kernel machine is decomposed into C components corresponding to each partition as follows:

$$f_c(\mathbf{x}) = \sum_{i=1}^{N_c} \alpha_i^c \, K_i^c(\mathbf{x} - \mathbf{x}_i^c)$$
$$f(\mathbf{x}) = \sum_{c=1}^{C} f_c(\mathbf{x}) \tag{3}$$

Next, we describe the parametric approximation technique we propose to simplify the calculation of each component.

3. SEPARABLE MAXIMUM ENTROPY MODEL

Throughout this section, we focus on a particular $f_c(\mathbf{x})$. The data $\mathbf{x}$ is assumed to be n-dimensional real vectors. This function will be approximated as a scaled version of a probability density function for an n-dimensional random vector $\mathbf{x}$, which is a linear combination of n independent source random variables. The linear combination matrix $\mathbf{A}$ will be determined using independent component analysis and the source distributions will be approximated as exponential densities with polynomial exponents, whose coefficients are determined using an approximate maximum entropy solution.

3.1. Determining Independent Component Directions

It is clear that assuming a separable approximation to $f_c(\mathbf{x})$ in the canonical directions in the data space might lead to poor approximations, while considering a joint parametric model fitting approach might run into optimization complexities due to high dimensionality and nonlinearity of the problem. In most situations, the modes determined

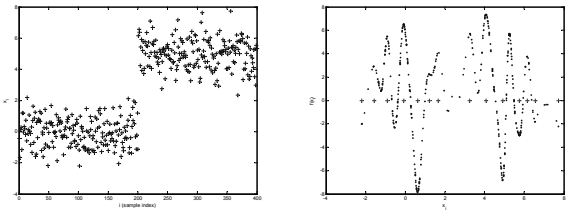

Figure 1. In (top) the one-dimensional training set x_i used in this illustration is plotted vs. the sample index. In (bottom) a *random* kernel machine is plotted for the given training data, where '+' indicates the fixed points of the iterations in (2).

using the clustering step would have shapes that conform to the linear independence assumption underlying ICA. Therefore, the proposed approach is a feasible and reasonable midpoint solution that eliminates difficulties associated with high dimensionality, without being as naïve as the separable-in-canonical-coordinates approach. Note, however, that there may be situations where the linear independence assumption is not valid.

In order to apply ICA theory, we first normalize the component function by normalizing the coefficients:

$$\beta_i^c = \left| \frac{\alpha_i^c}{\sum_i \alpha_i^c} \right| \tag{4}$$

Without loss of generality, we also subtract the mean of the data points ($\bar{\mathbf{x}} = \mathbf{x} - \sum_i \beta_i^c \mathbf{x}_i^c$) to obtain the zero-mean normalized *pdf* counterpart of $f_c(\mathbf{x})$:

$$h_c(\bar{\mathbf{x}}_i) = \sum_{i=1}^{N_c} \beta_i^c K_i^c(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_i^c) \tag{5}$$

Due to the linear independence assumption, we can approximate the joint pdf in (5) as a product of marginals

$$\hat{h}_c(\tilde{\mathbf{x}}) = \left|\mathbf{A}^{-1}\right| \prod_{d=1}^{M} h_{cd}(\tilde{x}_d) \tag{6}$$

where $\bar{\mathbf{x}} = \mathbf{A}\tilde{\mathbf{x}}$ and h_d is the d^{th} marginal of h:

$$h_{cd}(\tilde{x}_d) = \sum_i \beta_i^c K_{i,dd}^c(\tilde{x}_d - \tilde{x}_{i,d}^c) \tag{7}$$

One could optimize $\mathbf{A}$ in a variety of ways, one obvious but not straightforward way being the direct minimization of the Kullback-Leibler divergence between (5) and (6). Here, we propose drawing a large set of random samples from h and employing a standard sample-based ICA algorithm; in this paper we utilize a simple joint cumulant diagonalization algorithm based on generalized eigenvector decompositions [5].

Once $\mathbf{A}$ is determined using ICA, the function $f_c(\mathbf{x})$ can be decomposed into product of marginals using the rescaled version of (6). However, the marginals still require $O(N)$ kernel evaluations as shown in (7). Next, we show a convenient approximation for the marginals using the exponential family and the maximum entropy principle.

3.2. Maximum Entropy Marginal Models

For each h_{cd} in (7), we seek a parametric approximation that can be fit easily to the specific function. Maximum entropy principle [6] motivates the use of exponential parametric distributions and our previous work on maximum entropy density estimation provides an analytical approximate solution for the model parameters.

Maximum entropy principle: The maximum entropy distribution satisfying equality constraints for selected moments is characterized by the following optimization problem:

$$\max_{\hat{h}_d(\tilde{x})} -\int_{-\infty}^{\infty} \hat{h}_d(\tilde{x}_d)\log\hat{h}_d(\tilde{x}_d)\,d\tilde{x}_d \quad \text{subject to } E_{\hat{h}_d}[r_k(\tilde{x}_d)] = E_{h_d}[r_k(\tilde{x}_d)] \quad k=1,\ldots,m \tag{8}$$

where $r_k(.)$ are the nonlinear moments that define the constraints. The solution to this problem is given by [7]

$$\hat{h}_d(\tilde{x}_d) = C\exp\left(\sum_{k=1}^{m}\lambda_k r_k(\tilde{x}_d)\right) \tag{9}$$

Solving for the Lagrange multipliers involves integral equations or exponential functions, and is analytically intractable in general. However, an approximate analytical solution can be obtained [6]. Specifically, under the assumption that the true distribution is close to the maximum entropy distribution (which is expected to be the case in the specific case of compact modes of kernel machines), we can approximate the Lagrange multipliers by $\boldsymbol{\lambda} = -\boldsymbol{\Theta}^{-1}\boldsymbol{\eta}$, where $\boldsymbol{\lambda}=[\lambda_1,\ldots,\lambda_m]^T$, R_k is the integral of r_k and r'_k is the derivative, and the moment matrix-vector pair is:

$$\boldsymbol{\theta} = \begin{bmatrix} \vdots & \\ \cdots & E[R_i(\tilde{x}_d)r'_k(\tilde{x}_d)] & \cdots \\ \vdots & \end{bmatrix}, \quad \boldsymbol{\eta} = E\begin{bmatrix} \vdots \\ r_k(\tilde{x}_d) \\ \vdots \end{bmatrix} \tag{10}$$

The choice of $r_k(x)=x^k$ is particularly interesting for at least two reasons: (i) the corresponding exponential distribution can be regarded as a truncated Taylor series approximation to the logarithm of the actual distribution, (ii) the Lagrange multipliers are calculated using moments up to order $2m$ in a manner similar to moment matching. For this particular choice of constraint functions, the Lagrange multipliers can be approximated using sample estimates (with samples used in determining the ICA solution for **A**):

$$\theta_{ik} = \frac{k}{(i+1)}\frac{1}{N}\sum_{j=1}^{N}\tilde{x}_{j,d}^{i+k} \qquad \eta_k = \frac{1}{N}\sum_{j=1}^{N}\tilde{x}_{j,d}^{k} \tag{11}$$

Alternatively, these moments can be analytically computed for the particular kernel function. For the Gaussian kernel, for example, we have

$$\left\langle \tilde{x}_d^k \right\rangle_K = \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{\infty}\tilde{x}^k e^{-(\tilde{x}-\mu)^2/\left(2\sigma^2\right)}d\tilde{x} \tag{12}$$

The first 4 moments are

$$\begin{aligned} \left\langle \tilde{x}_{d,i}^1 \right\rangle_K &= \tilde{x}_i \\ \left\langle \tilde{x}_{d,i}^2 \right\rangle_K &= \tilde{x}_i^2 + \sigma_i^2 \\ \left\langle \tilde{x}_{d,i}^3 \right\rangle_K &= \tilde{x}_i\left(\tilde{x}_i^2 + 3\sigma_i^2\right) \\ \left\langle \tilde{x}_{d,i}^4 \right\rangle_K &= \tilde{x}_i^4 + 6\tilde{x}_i^2\sigma_i^2 + 3\sigma_i^4 \end{aligned} \tag{13}$$

Substituting the moments into (7) one can obtain the first 4 moments as

$$\begin{aligned} \boldsymbol{\eta}_1 &= \sum_i \beta_i\tilde{x}_i \\ \boldsymbol{\eta}_2 &= \sum_i \beta_i\tilde{x}_i^2 + \sigma_i^2 \\ \boldsymbol{\eta}_3 &= \sum_i \beta_i\tilde{x}_i\left(\tilde{x}_i^2 + 3\sigma_i^2\right) \\ \boldsymbol{\eta}_4 &= \sum_i \beta_i\left(\tilde{x}_i^4 + 6\tilde{x}_i^2\sigma_i^2 + 3\sigma_i^4\right) \end{aligned} \tag{14}$$

4. EXPERIMENTAL RESULTS

We present results on approximating discriminative kernel machine projections [8] on benchmark datasets.

Crescent dataset: This dataset consists of two crescent-shaped classes with a nonlinear class boundary. There are 150 samples in each class, which are generated by uniformly selecting the angle in a π-radian arc and perturbing the radius with Gaussian distributed random values. The centers of the semicircles describing the classes are selected such that there is a nonlinear separation boundary in between. This two-dimensional dataset and the performance comparison (ROC curves) of the original and approximate projection are shown in Figure 2. The area under the ROC curve of the original method is greater than that of its approximation. However, the approximation operates more accurately than the original in certain risk regimes. This might indicate an unintended regularization effect of the approximation procedure for certain datasets.

Wisconsin Breast Cancer Dataset & Ionosphere Dataset: Similar experiments were performed using the Wisconsin breast cancer dataset and the ionosphere dataset from the UCI database [ref]. The ROC curves for the original and approximate projections are shown in Figure 3.The approximation achieves performance levels similar to the original projection results for the Wisconsin breast cancer dataset, however, has severely degraded performance in the ionosphere dataset. Increasing the order of the exponential marginal approximations did not improve the performance of the approximation in this dataset. The most likely explanation to the failure of this

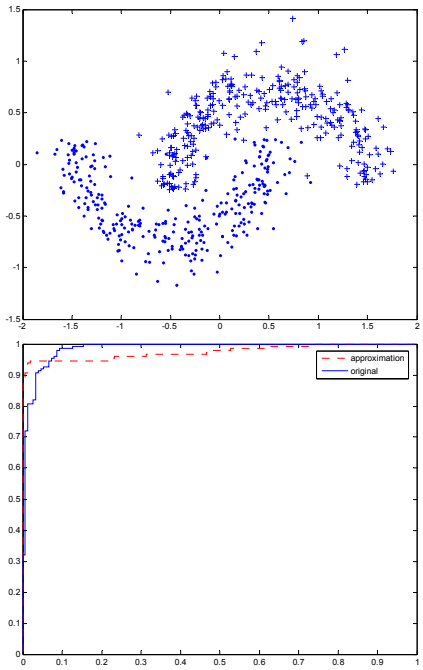

Figure 2. The crescent dataset with two classes indicated by two symbols is shown at the top. The ROC curves for the original (solid) kernel machine and the approximation (dashed) are shown at the bottom.

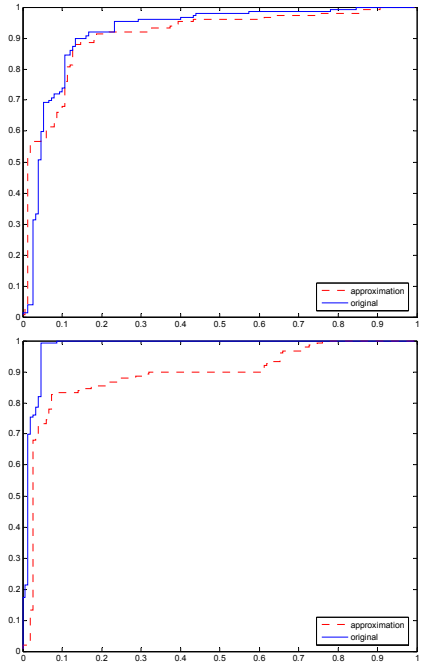

Figure 3. ROC curves of the original (solid) and approximate (dashed) kernel machines for the Wisconsin breast cancer dataset (top) and the ionosphere dataset (bottom).

method is that the linear independence assumption is violated for the modes of the kernel machine generated for this dataset.

5. DISCUSSION

The main drawback of the kernel machines is the high computational load in the testing phase. While parametric methods only need to calculate a few parameters, kernel machines need the evaluation of N kernel functions. This also means all training samples must be stored in memory. Currently available methods to overcome these drawbacks are based on sampling in the input feature space to throw away the *less informative* or some of the similar training samples. In this paper, we presented a *downsampling* approach that not only considers the density of samples but also their contribution to the function in their neighborhood. The function is partitioned into disjoint modes, which are then approximated by a linear combination of exponential marginal distributions. Experiments on benchmark datasets indicate mostly acceptable performance degredation, but severe degredation might also be possible when certain assumptions do not hold. Our future work will be focused on comparing the efficiency of the proposed algorithm with input feature space based approximation algorithms and relaxing the linearity assumption in the ICA step.

6. ACKNOWLEDGEMENTS

Authors would like to thank to Todd Leen and Tim Sheard for valuable discussions. This work was partially supported by NSF grant ECS-0524835.

7. REFERENCES

[1] B. Scholkopf, A. Smola, K.R. Muller, “Nonlinear Component Analysis as a Kernel Eigenvalue Problem,” Neural Computation, vol. 10, pp. 1299-1319, 1998.

[2] G. Baudat, F. Anouar, “Generalized Discriminant Analysis Using a Kernel Approach,” Neural Computation, vol. 12, pp. 2385-2404, 2000.

[3] D. Comaniciu, P. Meer, “Mean Shift: A Robust Approach Toward Feature Space Analysis,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603-619, 2002.

[4] M.A. Carreira-Perpiñán, C.K.I. Williams, “On the Number of Modes of a Gaussian Mixture,” Scale-Space Methods in Computer Vision, pp. 625-640, LNCS vol. 2695, Springer-Verlag, 2003.

[5] L. Parra, P. Sajda, “Blind Source Separation via Generalized Eigenvalue Decomposition”, Journal of Machine Learning Research, vol. 4, pp. 1261-1269, 2003.

[6] D. Erdogmus, K.E. Hild II, Y.N. Rao, J.C. Principe, “Minimax Mutual Information Approach for Independent Components Analysis,” Neural Computation, vol. 16, no. 6, pp. 1235-1252, 2004.

[7] E.T. Jaynes, “Information Theory and Statistical Mechanics,” Physical Review, vol. 106, pp. 620-630, 1957.

[8] U. Ozertem, D. Erdogmus, R. Jenssen, "Spectral Feature Projections That Maximize Shannon Mutual Information with Class Labels" to appear in Pattern Recognition, 2006.

[9] http://www.ics.uci.edu/~mlearn/MLRepository.html