Maximum Entropy PDF Projection: A Review

Paul M. Baggenstoss
Fraunhofer FKIE, Fraunhoferstrasse 20, 53343 Wachtberg, Germany
+49-228-9435-150, Email: p.m.baggenstoss@ieee.org

***Abstract*—We review maximum entropy (MaxEnt) PDF projection, a method with wide potential applications in statistical inference. The method constructs a sampling distribution for a high-dimensional vector $\mathbf{x}$ based on knowing the sampling distribution $p(\mathbf{z})$ of a lower-dimensional feature $\mathbf{z} = T(\mathbf{x})$. Under mild conditions, the distribution $p(\mathbf{x})$ having highest possible entropy among all distributions consistent with $p(\mathbf{z})$ may be readily found. Furthermore, the MaxEnt $p(\mathbf{x})$ may be sampled, making the approach useful in Monte Carlo methods. We review the theorem and present a case study in model order selection and classification for handwritten character recognition.**

I. Problem Statement

Given data $\mathbf{x} \in \mathcal{R}^N$, if we want to make statistical inference about some parameter θ using classical Bayesian methods, we require the probability density function (PDF) $p(\mathbf{x}|\theta)$. If not known, the PDF must be estimated, but the high dimension of $\mathbf{x}$ makes PDF estimation impractical. A solution is to extract one or more low-dimensional features $\mathbf{z}_k = T_k(\mathbf{x})$, $1 \le k \le K$, where $\mathbf{z}_k \in \mathcal{R}^{D_k}$, and $D_k << N$. Then, the feature PDFs $p(\mathbf{z}_k|\theta)$ can be estimated and used in place of $p(\mathbf{x}|\theta)$. But, how do we know which of the K features are best?

In existing methods, the effectiveness of feature $\mathbf{z}_k$ can only be measured in relation to other parameter points by comparing the PDF $p(\mathbf{z}_k|\theta_m)$ and $p(\mathbf{z}_k|\theta_l)$, $l \neq m$, for example in statistical trials measuring classification performance. Furthermore, this effectiveness may only apply for the chosen parameter points, desirable only in special *closed* classification problems, such as recognition of a fixed set of M characters. This is an example of *multilateral* feature selection, since it requires the choice of alternative parameter values to evaluate the features, and the result may only apply to that choice.

We would prefer a *unilateral* feature evaluation method that quantifies the effectiveness of feature $\mathbf{z}_k$ at a given parameter point θ_m alone (i.e. when considering just a single data class). This should greatly improve the generalizability of any classifier we construct. But what would be a criterion for such a search? This is related to the *model selection* problem [1]. In model selection or model order selection, we have some parametric form(s) of the data PDF $p(\mathbf{x}|k)$ depending on a model indexed by k. Since models with higher complexity tend to fit the data better, one seeks to balance models with higher complexity with some *penalty* function $f(k)$ such that the penalized likelihood $L(k) = p(\mathbf{x}|k) - f(k)$ reaches a maximum at the "best" value k. This general approach has two disadvantages. First, these penalized likelihood functions are based on asymptotic approximations, so do not result in PDFs, which limits the usefulness. Second, reasonable parametric models for the data may not exist. In the problem we discuss later, optical character recognition (OCR), what is a reasonable parameteric model for handwritten characters "3"? As we shall see, throwing general-purpose kernel-based data fitting models (Gaussian mixtures) at the data has its limitations. While we may not know a good parametric model for the data, we may know a good feature reduction that preserves the critical information. MaxEnt PDF projection converts this knowledge into a kind of parametric model.

II. PDF Projection

The method of PDF projection [2] provides a potential *unilateral* feature selection strategy. Let's assume that we know or have estimated the feature PDF $p(\mathbf{z}_k|\theta_m)$. Let $\{\mathcal{P}_x : T_k, p(\mathbf{z}_k|\theta_m)\}$ refer to the set of all PDFs $p(\mathbf{x})$ that are *consistent* with $p(\mathbf{z}_k|\theta_m)$. In other words, if $p(\mathbf{x}) \in \{\mathcal{P}_x : T_k, p(\mathbf{z}_k|\theta_m)\}$, then if samples of $p(\mathbf{x})$ are drawn and converted to features using $T_k(\mathbf{x})$, they will have precisely PDF $p(\mathbf{z}_k|\theta_m)$. The method of PDF projection [3], [4] finds a member of $\{\mathcal{P}_x : T_k, p(\mathbf{z}_k|\theta_m)\}$ based on choosing a reference distribution $p(\mathbf{x}|H_0)$.

We now state the main theorem from PDF projection. Let $\mathbf{z}$ be an arbitrary feature vector $\mathbf{z} = T(\mathbf{x})$, mapping $\mathbf{x} \in \mathcal{X} \subset \mathcal{R}^N$, to $\mathbf{z} \in \mathcal{Z} \subset \mathcal{R}^D$, where $D << N$ and let $g(\mathbf{z})$ be some arbitrary feature PDF on $\mathcal{X}$. Specifically, suppose that for some reference distribution $p(\mathbf{x}|H_0)$ on $\mathcal{X}$, we can precisely derive the feature PDF $p(\mathbf{z}|H_0)$. The reference hypothesis H_0 a mathematical reference distribution such as independent exponential or Gaussian noise and not be confused with the noise-only condition or any other other real data condition. The PDF projection theorem states that the function

$$G(\mathbf{x}; H_0, T, g) = \frac{p(\mathbf{x}|H_0)}{p(T(\mathbf{x})|H_0)} g(T(\mathbf{x})) = \frac{p(\mathbf{x}|H_0)}{p(\mathbf{z}|H_0)} g(\mathbf{z}) \quad (1)$$

is a PDF on $\mathcal{X}$ (it integrates to 1), and is a member of $\{\mathcal{P}_x : T, g(\mathbf{z})\}$. We call this PDF a *projected* PDF since it projects the PDF $g(\mathbf{z})$ defined on on $\mathcal{Z}$ onto the higer-dimensional space $\mathcal{X}$.

A strategy for feature selection would be to find projected PDFs $p_k(\mathbf{x}) \in \{\mathcal{P}_x : T_k, p(\mathbf{z}_k|\theta_m)\}$ for each k, given by

$$G(\mathbf{x}; H_0, T_k, p(\mathbf{z}_k|\theta_m)) = \frac{p(\mathbf{x}|H_0)}{p(\mathbf{z}_k|H_0)} p(\mathbf{z}_k|\theta_m), \quad (2)$$

then select that feature $\mathbf{z}_k$ that provides the highest average log-likelihood for some training set $\mathbf{x}_o$, $1 \le o \le O$,

$$L_k = \sum_{o=1}^{O} \log p_k(\mathbf{x}_o).$$

The flaw in this method is that the set $\{\mathcal{P}_x : T_k, p(\mathbf{z}_k|\theta_m)\}$ is infinitely large so we are not guaranteed that the choice of projected PDF at each k is fair, especially since different feature transformations $T_k(\mathbf{x})$, $T_l(\mathbf{x})$ may encompass completely different feature extraction methods.

It would be better if we could insure that for each k, the projected PDF could not only represent our knowledge of the data through higher average log-likelihood, but could also express our ignorance - through highest entropy. Entropy is just the negative of the theoretical average log-likelihood. The minimax solution would be to choose projected PDF for a given k for *minimum* theoretical log-likelihood (maximum entropy), then *maximizing* the average log-likelihood over k.

III. MaxEnt PDF Projection

Our goal is now to maximize the entropy of (1) over H_0. The entropy is given by

$$H_G = -\int_{\mathbf{x}} \log G(\mathbf{x}; H_0, T, g)\ G(\mathbf{x}; H_0, T, g)\ \mathrm{d}\mathbf{x}.$$

It can be shown (See [2], equation 8) that this can be expanded as follows:

$$H_G = H_g + \int_{\mathbf{z}} H_{\mu|z;H_0}\ g(\mathbf{z})\ \mathrm{d}\mathbf{z} \tag{3}$$

where the entropy of g is $H_g = -\int_{\mathbf{z}} \log g(\mathbf{z})\ g(\mathbf{z})\ \mathrm{d}\mathbf{z}$, and the manifold entropy is

$$H_{\mu|z;H_0} = -\int_{\mathbf{x}\in\mathcal{M}(z;T)} \log \mu(\mathbf{x}|\mathbf{z}; H_0)\ \mu(\mathbf{x}|\mathbf{z}; H_0)\ \mathrm{d}\mathbf{x},$$

where the manifold $\mathcal{M}(z;T)$ is defined as

$$\mathcal{M}(\mathbf{z}) = \{\mathbf{x} : T(\mathbf{x}) = \mathbf{z}\},$$

and $\mu(\mathbf{x}|\mathbf{z}; H_0)$ is the manifold distribution, the same as the posterior distribution of $\mathbf{x}$ given $\mathbf{z}$ under the prior $p(\mathbf{x}|H_0)$. Since $\mathbf{z}$ is deterministically derived from $\mathbf{x}$, the posterior is not proper. The manifold distribution has all its probability mass on the manifold, and is proportional to $p(\mathbf{x}|H_0)$ on the manifold. Since we have no information about the manifold distribution, the maximum entropy principle says that $\mu(\mathbf{x}|\mathbf{z}; H_0)$ should be the uniform distribution. Therefore, to maximize the entropy, we could use the uniform refrence hypothesis $p(\mathbf{x}|H_0) = 1$, and this would insure that $\mu(\mathbf{x}|\mathbf{z}; H_0)$ is uniform. This is only possible, however, if $\mathcal{X}$ is bounded, otherwise $p(\mathbf{x}|H_0)$ will be an improper distribution.

When $\mathcal{X}$ is unbounded, we have two problems, (a) we cannot use the uniform reference hypotheses which is improper, so we cannot insure that $\mu(\mathbf{x}|\mathbf{z}; H_0)$ is uniform, and (b) a given manifold $\mathcal{M}(\mathbf{z})$ may unbounded, i.e. there may be members of $\mathcal{M}(\mathbf{z})$ where $\|\mathbf{x}\| = \infty$. The solution to this dilemma was first proposed in [2]. Let there exist a function f such that

$$f(\mathbf{z}) = f(T(\mathbf{x})) = \|\mathbf{x}\| \tag{4}$$

for some norm $\|\mathbf{x}\|$ valid in $\mathcal{X}$. Then, if the reference distribution can be written in the form

$$p(\mathbf{x}|H_0) = h(T(\mathbf{x})) \tag{5}$$

for some some function h, then the projected PDF (1) is the member of $\{\mathcal{P} : T, g\}$ with highest entropy. Clearly (5) must be constant on $\mathcal{M}(z;T)$ since $T(\mathbf{x})$ is fixed. Therefore, $\mu(\mathbf{x}|\mathbf{z}; H_0)$ is the uniform distribution. Also, since by (4), as long as $\mathbf{z}$ is finite-valued, $\|\mathbf{x}\|$ is constrained to a constant on the manifold, so the manifold itself must be a compact set and the uniform distribution is the MaxEnt distribution on the manifold [5].

The most straight-forward way to achieve both (4) and (5) simultaneously is to choose an apropriate *energy statistic* $t(\mathbf{x})$. By this, we mean that $\|\mathbf{x}\|$ can be computed from $t(\mathbf{x})$ and $t(\mathbf{x})$ can be computed from $T(\mathbf{x})$. We can then choose a reference distribution in the exponential family

$$p(\mathbf{x}|H_0) = C \exp\left\{-|t(\mathbf{x})/a|^p\right\}.$$

This family includes standard normal and exponential distributions. For example, by including energy statistic

$$t_2(\mathbf{x}) = \sum_{n=1}^{N} x_n^2$$

, which leads to the 2-norm, we can choose H_0 to be the canonical Gaussian distribution

$$p(\mathbf{x}|H_0) = \prod_{n=1}^{N} (2\pi)^{-1/2} e^{-x_n^2} = (2\pi)^{-N/2} e^{-t_2(\mathbf{x})/2}.$$

For positive-valued $\mathbf{x}$ and inclusion of energy statistic

$$t_1(\mathbf{x}) = \sum_{n=1}^{N} x_n$$

, we can choose H_0 to be the canonical exponential distribution

$$p(\mathbf{x}|H_0) = \prod_{n=1}^{N} e^{-x_n} = e^{-t_1(\mathbf{x})}.$$

And, of course, for $\mathbf{x}$ with elements in the range $[0,\ 1]$, the we can choose H_0 to be the uniform distribution $p(\mathbf{x}|H_0) = 1$.

A. The Chain Rule

In practice, feature extraction can take the form of multiple stages of processing. At the output of a long chain, it may be difficult or impossible to carry out the necessary derivation of $p(\mathbf{z}|H_0)$, which is the feature PDF under the assumption that the input data is distributed according to the canonical reference hypothesis H_0. The Chain-rule makes constructing a projected PDF based on multi-stage feature extraction much easier. The chain $\mathbf{y} = T_y(\mathbf{x}),\quad \mathbf{w} = T_w(\mathbf{y}),\quad \mathbf{z} = T_z(\mathbf{w})$, suggests the chain-rule form of (1),

$$G(\mathbf{x}) = \left[\frac{p(\mathbf{x}|H_{0x})}{p(\mathbf{y}|H_{0x})}\right]\left[\frac{p(\mathbf{y}|H_{0y})}{p(\mathbf{w}|H_{0y})}\right]\left[\frac{p(\mathbf{w}|H_{0w})}{p(\mathbf{z}|H_{0w})}\right] g(\mathbf{z}), \tag{6}$$

where H_{0x}, H_{0y}, H_{0w} are stage-dependent statistical hypotheses. The reference hypothesis at each stage can be set to a canonical reference hypothesis, making it easier to derive the PDF at the output of that stage. Interestingly, a projected PDF constructed with the chain-rule can also be thought of

as equation (1), but with a compound reference hypothesis. Thus, (6) can be seen as

$$G(\mathbf{x}) = \frac{p(\mathbf{x}|H_0)}{p(\mathbf{z}|H_0)} g(\mathbf{z}),$$

where

$$p(\mathbf{x}|H_0) = \left[\frac{p(\mathbf{x}|H_{0x})}{p(\mathbf{y}|H_{0x})}\right] \left[\frac{p(\mathbf{y}|H_{0y})}{p(\mathbf{w}|H_{0y})}\right] p(\mathbf{w}|H_{0w}),$$

which is, itself a PDF projection.

The chain rule also suggests an elegant modular software framework for a feature extraction chain: `[y,J]=stage1(x,J);` , then `[w,J]=stage2(y,0);` , etc., where variable `J` accumulates the log-PDF ratios $\log\{p(\mathbf{x}|H_{0x})/p(\mathbf{y}|H_{0x})\}$, $\log\{p(\mathbf{y}|H_{0y})/p(\mathbf{w}|H_{0y})\}$, ... Both PDF projection and maximum entropy PDF projection extend recursively using the chain-rule. In other words, (6) is a PDF (it integrates to 1), it is a member of the class of PDFs that generate $g(\mathbf{z})$, and if the conditions for maximum entropy in Section III hold individually at each stage, then it is the maximum entropy member. When drawing samples from $G(\mathbf{x}; g)$, we work backward through the chain. We first draw a sample $\mathbf{z}$ according to $g(\mathbf{z})$, then draw a sample $\mathbf{z}$ uniformly distributed on the set $\mathbf{w} : T_z(\mathbf{w}) = \mathbf{z}$, etc.

IV. Application: OCR

We now apply MaxEnt PDF projection to choosing features for optical character recognition (OCR), and later for combining multiple projected PDFs.

A. Data description

The MNIST OCR data [6] set consists of ten handwritten digits 0-9 divided into two sub-corpora: the training sub-corpus with 6000 training samples of each digit, and the testing sub-corpus with 1000 testing samples of each digit. To limit the processing, we inverted the roles training and testing sub-corpora (we used the testing sub-corpus for training). We also downsampled the 28×28 images 2:1 to 14×14 and limited our experiments to three digits "3", "8", and "9". We therefore had a total of about 3000 training samples of dimension $N = 14 \times 14 = 196$. The data is positive-valued in the range [0,1].

B. Benchmark Classifiers

To provide a performance benchmark, we applied two widely-used methods, support vector machine (SVM) [7] and a multi-layer perceptron (MLP). The results are listed in Table I in order of decreasing error. The best performance of 472 errors (2.63%) was obtained with the SVM classifer using polynomial kernel [7].

C. Direct Gaussian Mixture

To begin our quest for features, we started by modeling the data directly, with no feature reduction, by a Gaussian mixture with full covariance matrices. Because of the high dimension, we added a positive constant to the covariance matrix diagonal to prevent ill-conditioning.

Classifier	Pre-Proc	Total Errors (percent)
SVM (radial kernel)	atanh	11800 (65.8%)
SVM (radial kernel)	none	1575 (8.78%)
SVM (linear kernel)	atanh	965 (5.38%)
MLP (no hiddel layers)	none	947 (5.28%)
SVM (linear kernel)	none	870 (4.85%)
MLP (9 hiddel layers)	none	754 (4.20%)
SVM (polynomial kernel)	atanh	667 (3.72%)
MLP (12 hiddel layers)	none	640 (3.56%)
MLP (24 hiddel layers)	none	594 (3.31%)
MLP (50 hiddel layers)	none	514 (2.86%)
SVM (polynomial kernel)	none	472 (2.63%)

TABLE I
BENCHMARK RESULTS

We measured total self-approximation likelihood using only the *training data* (the MNIST "testing" sub-corpus). We used three-fold cross-validation by estimating the Gaussian mixture PDF on two-thirds of the data, and evaluating the log-likelihood on the remaining one-third, then repeated for each of the three possible sub-divisions. This was done independently for each of the three classes, and the results summed to obtain a single number. The testing data (the MNIST "training" sub-corpus) was reserved to measure classification errors after PDF estimation on the full training set. We did this to demonstrate the potential of predicting classifier performance just based on total log-likelihood.

Figure 1 (curve labeled "DIRECT") shows both total self-approximation likelihood and classifier error as a function of diagonal loading constant. Note that the log-likelihood peaks at about 0.4, which reasonably predicts a good value of diagonal loading variance for classification. But, one is not always so fortunate. It is possible that the likelihood peak occurs at a place that gives poor classification performance. When this occurs, it may indicate that likelihood is either "chasing" information irrelevant to the classifcation decision (nuisance information), or the model is extremely mismatched to the data. An example of nuisance information is vocal pitch frequency in speech recognition, which tends to carry little information about the spoken word, but knowing it can make an enormous difference in model fitting to the time-series. While nuisance information is difficult to identify quantitatively by the unilateral approach (see Section I), one can unilaterally eliminate model mismatch by choosing models that improve likelihood fitting (but only if cross-validated with a separate testing data set). We will demonstrate this now.

In fact, the Gaussian mixture is very poorly suited to data constrained to the hypercube with values between 0 and 1. So, it makes sense to transform the data with a monotonic increasing function that maps the values in [0,1] to the real line. We tried two transformations: the "atanh" function $y = \mathrm{atanh}(1.997(x - .5))$, and the "unlogist" transformation $y = -\log(1/(x - 1))$, the inverse of the logistic (sigmoid) function. The constant 1.997 was used better to handle data near the boundaries $x = 0$ and $x = 1$ and was optimized for maximum total likelihood value. In order to compare

likelihood values before and after transforming, the Jacobian of the transformation was determined so that the PDF values can be referenced to the original data space. For example, for the "unlogist" transformation,

$$p(\mathbf{x}) = \left\{ \prod_{i=1}^{N} \frac{1}{x_i(1-x_i)} \right\} p(\mathbf{y}).$$

Figure 1 shows the results for two transformations and for "direct" PDF approximation as a function of diagonal loading constant. The average of seven trials is shown. In all cases, using a Gaussian mixture of 2 mixture components worked best, providing highest log-likelihood and lowest error. The

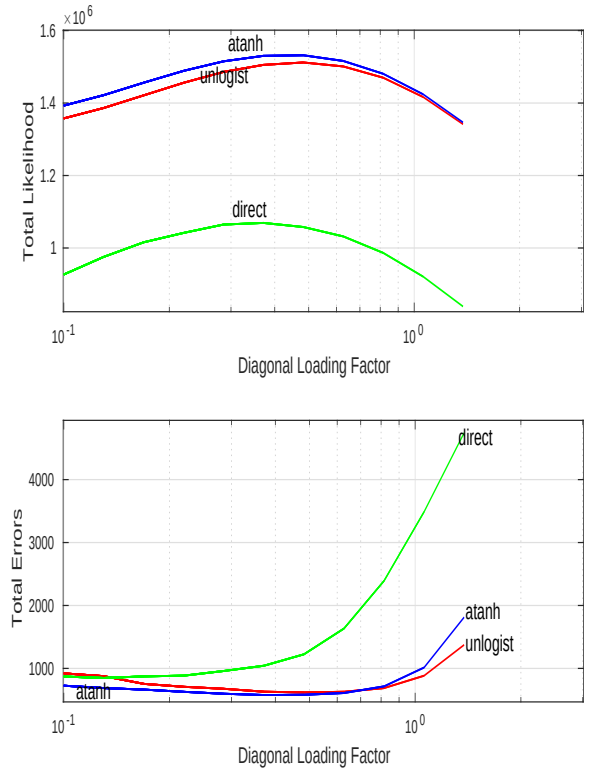

Fig. 1. Top: total log-likelihood, Bottom: total classification error, both as a function of diagonal loading constant.

"atanh" transformation was the best with a peak log-likelihood of 1.532×10^6, which improved upon the "direct" method by 4.6×10^5. To appreciate this number, if we divide by the number of test samples (3000), then divide by $N = 196$, it comes to a log-likelihood difference of .78 per dimension, or about a factor of 2 for each of the 169 dimensions! The best average errors (in 3 trials) were DIRECT: 850 (4.74%), UNLOGIST: 620(3.46%), and ATAHN: 575(3.21%). The log-likelihood curve provides a reasonable estimate of the best value of the diagonal loading variance, and also predicts which transformation works best. Now we ask if it is possible to improve the model fitting with PDF projection.

D. PDF Projection using PCA

Now that we have gotten the most out of the full-dimensional PDF estimation approach, which previously we described as just "throwing a general-purpose kernel-based data fitting model at the data", we now see what MaxEnt PDF projection can do. A widely-used dimension-reducing approach that preserves critical information is PCA analysis.

Let $\mathbf{V}_P$ be the $N \times P$ ortho-normal matrix derived from PCA of the training samples. Our $(P+1) \times 1$ feature vector was then

$$\mathbf{z}_P = \begin{bmatrix} \mathbf{V}_P'\mathbf{y} \\ \\ \mathbf{y}'\mathbf{y} - \mathbf{y}'\mathbf{V}_P\mathbf{V}_P'\mathbf{y} \end{bmatrix}, \tag{7}$$

where $\mathbf{y}$ is the output of the "atanh" transformation (see previous section), and the last component is the residual energy (noise subspace energy). The $t_2(\mathbf{y})$ energy statistic can be computed from $\mathbf{z}$, so the feature transformation meets the conditions for Maximum entropy PDF projection (see Section III). Computing the PDF projection for $\mathbf{z}_P$, i.e. equation (2), requires knowing $p(\mathbf{z}_P|H_0)$ under the canonical Gaussian distribution. The details are provided in [2], Section IV.C, on page 2821, where $\mathbf{V}_P$ takes the place of $\mathbf{A}$. The feature PDFs $p(\mathbf{z}_P|\theta_m)$, where m indexes the three classes $1 \leq m \leq 3$, were estimated by Gaussian mixture approximation using 3 mixture components and full covariance matrices and diagonal loading variance 0.3. We converted projected PDFs on $\mathbf{y}$ to PDFs on $\mathbf{x}$ using the Jacobian as explained in Section IV-C.

We conducted the analogous experiment to Figure 1, but using projected likelihood values and varying the subspace dimension P. Note that in keeping with strict data separation, we determined the feature PDFs $p(\mathbf{z}_P|\theta_m)$ using just the 2/3 of the data from each class reserved for training. We also determined the PCA matrix $\mathbf{V}_P$ using the same 2/3 of the data from all classes reserved for training. We measured the total projected log-likelihood on the remaining third. Figure 2 shows the result of this experiment as a function of subspace dimension P (curve labeled "PCA"). The curves show the average of seven independent experiments. The cross-validated log likelihood peaked at $P = 50$ with a value 1.574×10^6, an improvement of 54,000 over the full-dimensional Gaussian mixture PDF estimation method. If divided by the 3000 testing samples and the 196 dimensions, this can be appreciated as a log-likelihood difference of factor of .092, or a factor of 0.9 for each dimension of $\mathbf{x}$, still significant. The classification error had a minimum at about $P = 40$ of about 425 errors (2.37%). Again, the location of the log-likelohood peak gives a reasonable approximation to the best P for classification.

E. Class-dependent PDF Projection using PCA

In the last experiment, we used PDF projection to predict the performance of features in a classifier. But, PDF projection was not needed in the classification itself since a fixed feature transformation was used, so the only term in the projected PDF (2) that depended on the class assumption was the last one (the feature PDF). But, maximum entropy PDF projection

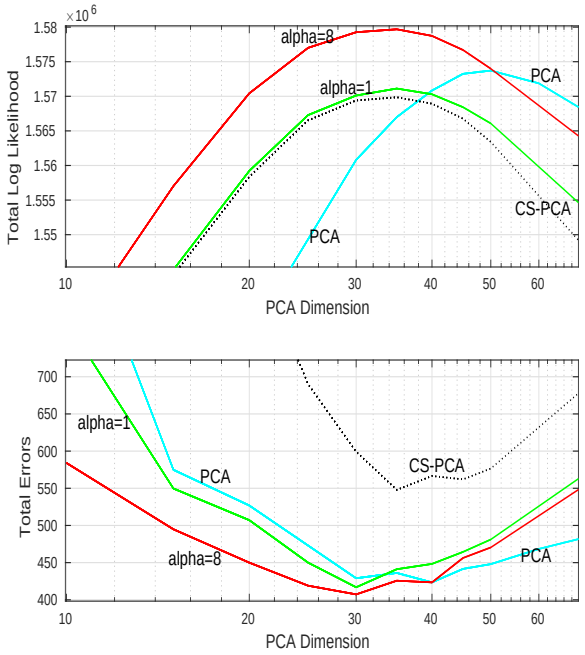

Fig. 2. Left: total projected log-likelihood (cross-validated) as a function of feature dimension - showing collapse above dimension 30. Right: classification error showing an inverse relationship to projected log-likelihood.

allows projected likelihood functions to be compared, even if they are constructed using different feature transformations. If we determine the PCA matrix $\mathbf{V}_P$ separately for each data class, we obtain a set of M feature transformations, $z_{P,m} = T_{P,m}(\mathbf{x})$, modeled on (7). Figure 2 shows the results (curve "CS-PCA"). Class-dependent PCA achieves the highest log-likelihood at a lower dimension ($P = 30$), but the highest likelihood is lower and the errors higher than for the common "PCA". We solve this problem in the next section.

F. Class-specific Model Mixture

The problem with class-dependent features has been previously explained, and arises from the inherent *one class - one feature* assumption in the classifier structure. While most testing samples of a class are best modeled by the corresponding feature, there will be some samples that will be best modeled by the features of another class. To solve this problem, we proposed the class specific model mixture [8], [9] in which we form a mixture of projected PDFs by expanding (2) in an additive mixture PDF:

$$\begin{aligned} \hat{p}(\mathbf{x}|\theta_m) &= \textstyle\sum_{l=1}^{M} w_{l,m}\, \hat{p}(\mathbf{x}|\theta_m, T_l) \\ &= \textstyle\sum_{l=1}^{M} w_{l,m}\, \frac{p(\mathbf{x}|H_0)}{p(\mathbf{z}_l|H_0)} \hat{p}(\mathbf{z}_l|\theta_m), \end{aligned} \tag{8}$$

where $\sum_l w_{l,m} = 1$. These weights are determined separately for each class assumption using test data. Results of classifying with (8) are shown in Figure 2 (curve "alpha=1"), and are compared with class specific PCA PDF projection ("CS-PCA"). There was a slight increase in log-likelihood and a slight reduction in error. This can be understood since the errors are few with respect to the total number of samples, so only a slight likelihood improvement can be expected.

G. Alpha Integration

We have previously described an improvement to the class-specific model mixture through alpha-intergration [9]. We let

$$\hat{p}(\mathbf{x}|\theta_m) = \frac{1}{C(\alpha)} \left(\sum_{l=1}^{M} w_l \left[\frac{p(\mathbf{x}|H_0)}{p(\mathbf{z}_l|H_0)} \hat{p}(\mathbf{z}_l|\theta_m) \right]^{1/\alpha} \right)^{\alpha}, \tag{9}$$

where the constant $C(\alpha)$ is such that

$$\int_{\mathbf{x}} \hat{p}(\mathbf{x}|H_m) = 1.$$

For classifying, the constant $C(\alpha)$ may be ignored and good results can be obtained [9]. But, for likelihood comparison, as is our goal in this paper, it cannot be ignored.

To obtain the constant, we use Monte Carlo integration, in which we carry out the integration of an arbitrary function $h(\mathbf{x})$ as

$$\int_{\mathbf{x}} h(\mathbf{x}) \mathrm{d}\mathbf{x} = \int_{\mathbf{x}} \frac{h(\mathbf{x})}{p(\mathbf{x})} p(\mathbf{x}) \mathrm{d}\mathbf{x} \sim \frac{1}{K} \sum_{k=1}^{K} \frac{h(\mathbf{x}_k)}{p(\mathbf{x}_k)},$$

where the approximation sums over K samples of $\mathbf{x}$ drawn from the "proposal" distribution $p(\mathbf{x})$. For the proposal distribution, we need a distribution that covers the region of support of the function to be integrated. Therefore, if we are integrating (9) for class m, we use (8) as the proposal distribution.

To draw samples of (8) from class m, we first choose a discrete random variable l from the discrete probability distribution $w_{l,m}$, $1 \le l \le M$ - This chooses the feature transformation. Next, we draw a feature $\mathbf{z}_l$ from the Gaussian mixture approximated distribution $p(\mathbf{z}_l|\theta_m)$. Finally, we draw a sample $\mathbf{y}$ from the level set $\{\mathbf{y} : T_l(\mathbf{y}) = \mathbf{z}_l\}$ (with a uniform distribution on the level set), then convert from $\mathbf{y}$ to $\mathbf{x}$ using the inverse of the "atanh" function, $x_i = \tanh(y_i)/1.997 + .5$, $1 \le i \le N$. The level-set sampling is accomplished as follows. Define the components of feature (7) as $\mathbf{z}_v = \mathbf{V}'\mathbf{x}$, $z_e = \mathbf{y}'\mathbf{y} - \mathbf{z}_v'\mathbf{z}_v$. Then, if $\mathbf{y}$ is on the level set, it must obey the two constraints

$$\mathbf{V}'\mathbf{y} = \mathbf{z}_v, \quad \mathbf{y}'\mathbf{B}'\mathbf{B}\mathbf{y} = z_e,$$

where $\mathbf{B}$ is the $N \times (N-P)$ orto-normal matrix spanning the space orthogonal to $\mathbf{V}$. We can span all solutions to the first constraint with an $(N-P) \times 1$ vector $\mathbf{u}$ as follows

$$\mathbf{y} = \mathbf{V}\mathbf{z}_v + \mathbf{B}\mathbf{u}.$$

To meet the second constraint, it follows that $\mathbf{u}'\mathbf{u} = z_e$. Therefore, $\mathbf{u}$ must be sampled uniformly on the sphere of radius $\sqrt{z_e}$. We therefore create $\mathbf{u}$ by forming an $(N-P) \times 1$ vector of independent Gaussian random variables, then normalizing it to have norm $\sqrt{z_e}$. Finally, we let $\mathbf{y} = \mathbf{V}\mathbf{z}_v + \mathbf{B}\mathbf{u}$.

Examples of samples of character "3" created in this way are shown in Figure 3 using a subspace dimension of $P = 25$.

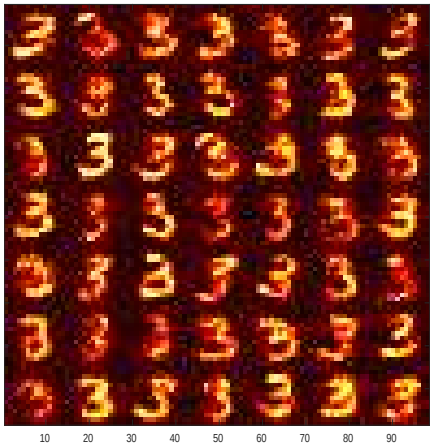

Fig. 3. 49 random samples of digit "3" drawn by maximum entropy synthesis with $P = 25$.

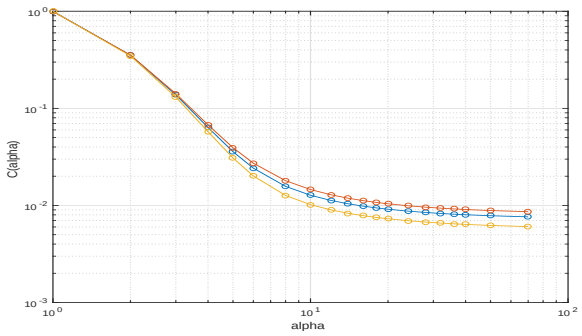

Fig. 4. Estimated value of $C(\alpha)$ as a function of α for the three classes at $P = 35$.

Figure 4 shows the estimated value of $C(\alpha)$ as a function of α for the three classes at $P = 35$. Note that $C(1) = 1$, as expected. In Figure 2 (curves "alpha=1" and "alpha=8") we see the result for $\alpha = 1$ and $\alpha = 8$. We experimented with several values of α and found that $\alpha = 8$ maximized log-likelihood. For $\alpha = 8$, the total log likelihood exceeded that for Common PCA (curve "PCA") and the classification error decreased from 424 to 407 (2.37% to 2.27%).

H. Hybrid Classifiers

By combining the best benchmark classifier (Section IV-B), the SVM, with the PDF projection classifiers, we can obtain performance exceeding either component. Using a simple additive rule, adding the log-likelihood scaled by a constant c to the output of the SVM, we obtain a simple hybrid classifier. Results are shown in Table II. The best result was 349 errors (1.94%). This compares favorably with the SVM alone: 472 errors (2.63%) or the best generative classifier: 407 errors (2.27%).

Classifier	P	Factor c	Total Errors (percent)
PCA	40	3.0	353 (1.97%)
CS-PCA ($\alpha = 8$)	30	3.0	349 (1.94%)

TABLE II
RESULTS OF COMBINING PDF-PROJECTION-BASED CLASSIFIERS WITH SVM.

V. Conclusions and Future Work

We have introduced maximum entropy PDF projection and have applied it to the problem of feature selection and classification. We have shown the ability to predict classifier performance based on average log-likelihood with cross-validation. We have also demonstrated that we can create generative classifiers that surpass the performance of popular discriminative classifiers.

Generative methods, which in lhe last decades took a backseat to popular discriminative methods, are now seeing a re-birth, for example, in a form of Bayesian belief network called deep belief network (DBN) [10], [11]. We see the potential for maximum entropy PDF projection to add to this trend and to become useful to analyze neural networks.

References

[1] K. P. Burnham and D. R. Anderson, *Model Selection and Multimodel Inference, Second Edition*. Springer, 2002.

[2] P. M. Baggenstoss, "Maximum entropy pdf design using feature density constraints: Applications in signal processing," *IEEE Trans. Signal Processing*, vol. 63, no. 11, 2015.

[3] ——, "A theoretically optimum approach to classification using class-specific features." *Proceedings of ICPR, Barcelona*, 2000.

[4] ——, "The PDF projection theorem and the class-specific method," *IEEE Trans Signal Processing*, pp. 672–685, March 2003.

[5] J. N. Kapur, *Maximum Entropy Models in Science and Engineering*. Wiley (Eastern), 1993.

[6] J. LeCun, "Mnist database," 2014. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[7] T. Joachims, "Svm-light toolkit," 2014. [Online]. Available: http://svmlight.joachims.org/

[8] P. M. Baggenstoss, "Optimal detection and classification of diverse short-duration signals," in *Proceedings of the International Conference on Clould Engineering*, Boston, MA, 2014, pp. 534–539.

[9] ——, "Class-specific model mixtures for the classification of acoustic time-series," *IEEE Trans. AES (accepted)*, 2016.

[10] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," 2006.

[11] G. E. Hinton, "2007 nips tutorial on deep belief nets," 2007. [Online]. Available: https://www.cs.toronto.edu/ hinton/nipstutorial/nipstut3.pdf