Maximum Entropy Feature Fusion

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

***Abstract*—We review recent theoretical results in maximum entropy (MaxEnt) PDF projection that provide a theoretical framework for fusing the information from multiple features for the purpose of general statistical inference. Given a high-dimensional input data vector x, and several dimension-reducing feature transformations $z_i = T_i(x)$, we consider the problem of estimating the probability density function (PDF) of x by fusing the information in the various features. When the PDF of one feature $p(z_i)$ is known or has been estimated, the PDF $p_i(x)$ that has maximum entropy among all PDFs consistent with $p(z_i)$ can be constructed. This is called the maximum entropy projected PDFs and can serve as a generative models from which random samples can be drawn. The information from all the features can be fused into a common classifier structure either by testing each hypothesis with a different feature, or by combining the various projected PDFs in a mixture PDFs. We review related theoretical and experimental results and provide a simulated classification experiment to highlight the potential of the method.**

I. Introduction

We are concerned with the problem of making statistical inferences from data. In classical decision theory [1], [2], we perform statistical inference based on the probability density functions (PDFs). Given some data $\mathbf{x} \in \mathcal{R}^N$, and M possible class decisions, the *most likely* class H_m is that which maximizes

$$\arg\max_m p(\mathbf{x}|H_m)\, p(H_m), \quad 1 \leq m \leq M.$$

This is the origin of the so-called Bayesian or *generative* methods - refering to the generative capability of the statistical models. Generative methods worked well if $p(\mathbf{x}|H_m)$ was known up to a few unknown parameters, or for N small. But for high-dimensional $\mathbf{x}$, a fixed amount of training data, and with no knowledge of the parametric form of the PDF, they suffered from the dimensionality curse. Later, classical learning theory [3], argued that classical decision theory needlessly solves the more general problem of PDF estimation, so it is better to estimate the posterior probabilities $p(H_m|\mathbf{x})$ directly. This idea gave rise to the so-called discriminative methods, known today as perceptrons, support vector machines, and deep networks. One of the problems of disciminative methods is that they do not generalize well. The decision function that declares "class A" only works well in the context of the chosen alternative classes "class B", "class C", etc. Furthermore, determining exactly what has been learned about an individual class is mostly meaningless for a discriminatlve model. The generative methods, which took a back-seat to the popular discriminative methods, are now making a come-back. There has been increased interest in generative models based on sigmoid belief networks (SBN) and deep belief networks (DBN) [4], [5]. One, less widely known method that belongs to this resurgence of generative models is the method of PDF projection [6]. Unlike other generative models which provide an explicit model for $p(\mathbf{x}|H_m)$, PDF projection finds the *implicit* model corresponding to an explicit dimension-reducing transformation. Therefore, PDF projection may be the method of choice when for practical reasons, one is forced to a lower-dimensional feature space.

II. Problem Definition

Let us assume that we have some high-dimensional data $\mathbf{x} \in \mathcal{R}^N$ about which we would like to make inferences. Let us also assume that estimating the PDFs $p(\mathbf{x}|H_m)$ is either impossible since we cannot observe $\mathbf{x}$ directly, or impractical because the dimension is too large. In place of $\mathbf{x}$, we have L feature vectors of the form $\mathbf{z}_i = T_i(\mathbf{x})$, $1 \leq i \leq L$. Each feature transformation $T_i(\mathbf{x})$ can be scalar or multi-dimensional, but must be dimension-reducing. Each feature vector can be considered as a different "view" of the raw data $\mathbf{x}$. Examples of this problem occur in many fields including remote sensing, compressive sensing, classification, and machine learning. Since we have assumed that it is impossible or impractical to work directly with $\mathbf{x}$, we are forced to work with just the features. The most straight-forward approach would be to form the super-vector from the union of the features $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2 \ldots \mathbf{z}_L\}$, then apply well-known discriminative approaches such as neural networks, support vector machines, and deep learning to approximate $p(H_m|\mathbf{Z})$. While discriminative approaches have many advantages, they do not generalize well unless they are trained on very large and diverse training sets. Furthermore, they are black boxes that are difficult to interrogate (visualize, validate) and often do not provide linear measures of confidence in the decision.

The generative approaches require estimating the PDFs. Since, we assume that the estimation of $p(\mathbf{x}|H_m)$ is impossible or impractical, we are forced to cast the problem in terms of the features. Estimating the PDFs of the super-vector $p(\mathbf{Z}|H_m)$ is problematic when the dimension of $\mathbf{Z}$ is large and we encounter the curse of dimensionality [7]. Generative approaches generally do not work as well as discriminative approaches at higher feature dimensions [3]. This leaves dimension-reducing methods such as feature selection, linear regression, principal component analysis (PCA), Fisher discriminant analysis (FDA) and manifold learning [8], [9]. Dimension-reduction approaches are just compromises between the two evils of lost information and PDF approximation error.

Once a classifier works with ad-hoc feature vectors, neither the discriminative nor the generative approaches can claim optimality with respect to the original data $\mathbf{x}$. We therefore seek a method to make statistical inference about the raw data, in the raw data domain, but based on low-dimensional measurements (features). It is well known that if the moments of $\mathbf{x}$ can be measured and specified, (mean, variance, skew, kurtosis, etc,) then there are well-established methods of designing the PDFs that have maximum entropy among all PDFs meeting the moments constraints [10], [11], [12]. In machine learning and related fields, it is rare to use moments, one typically uses arbitrary features. But, surprisingly, there have been very few attempts to extend the method of moments to feature constraints. One method by Basu [13] is based on the marginal distributions of the individual features (elements of the feature vector). In a previous publication [14], we succeeded in extending the method of moments to features PDFs for arbitrary feature transformations. In other words, instead of specifying the moments of the PDF, we specify the PDFs of feature vectors that are derived from $\mathbf{x}$, then obtain the maximum entropy PDF $p(\mathbf{x})$ among all PDFs that generate the specified feature PDFs. This constitutes a new generative classifier approach called maximum entropy PDF projection (MEPP) that rigorously includes the raw data in the theoretical framework. The method therefore requires a careful analysis of the feature extraction transformations.

III. Theoretical Results

The method of maximum entropy (MaxEnt) PDF projection (MEPP) was introduced in 2015 [14] and is an extension of PDF projection which was introduced on 2000 [15], [6]. While other generative approaches consider $p(\mathbf{x})$ unknowable, or impractical to estimate, MEPP starts with a simple question" "Given I have estimated $p(\mathbf{z}_i|H_m)$, the PDF of a feature $\mathbf{z}_i = T_i(\mathbf{x})$, under some hypothesis H_m, what does this tell me about $p(\mathbf{x}|H_m)$?" In other words, what is a reasonable estimate of $p(\mathbf{x}|H_m)$ given I know $p(\mathbf{z}_i|H_m)$? The concept is illustrated in Figure 1. By measuring the distribution of two features, we obtain two differing views or "opinions" about the distribution of the raw data $\mathbf{x}$. We now ask "is there a way to formalize the process of infering the distribution of $\mathbf{x}$ when one knows the distribution of features?", and "What is the optimal way to solve this inference problem?"

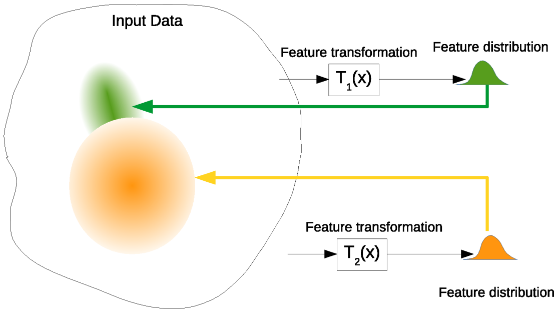

Fig. 1. Notional diagram of PDF projection: inference about the distribution of the input data based on the distribution of features.

We will see that by answering these questions, a world of possibilities open up. It is possible to perform statistical inference in the raw data domain, while only needing to measure low-dimensional features. Instead of being bound to one feature, one could use multiple features, even a different feature to test each hypothesis H_m. Or, one could form mixture distributions for a given hypothesis by employing several features. These questions are answered by the method of PDF projection and its offspring MEPP.

A. PDF Projection

Let $\mathbf{z}$ be an arbitrary feature vector $\mathbf{z} = T(\mathbf{x})$ and let $g(\mathbf{z})$ be some arbitrary feature PDF. Let $\mathcal{P}(\mathbf{x})\{g(\mathbf{z})\}$ be the class of PDFs $p(\mathbf{x})$ which generate $g(\mathbf{z})$. In other words, if $p(\mathbf{x}) \in \mathcal{P}(\mathbf{x})\{g(\mathbf{z})\}$, then if we draw samples $\mathbf{x}$ from $p(\mathbf{x})$, and pass them through $T(\mathbf{x})$, the resulting features will be samples of $g(\mathbf{x})$. PDF projection [15], [6] is the method of finding a member of $\mathcal{P}(\mathbf{x})\{g(\mathbf{z})\}$ by defining a reference hypothesis H_0. All unique members of $\mathcal{P}(\mathbf{x})\{g(\mathbf{z})\}$ correspond to a unique reference hypothesis H_0. The reference hypothesis H_0 is an assumed distribution for $\mathbf{x}$ but should not be confused with the noise-only condition or any other other real data condition. It is a mathematical reference distribution such as independent Gaussian noise. The PDF projection theorem [6] (PPT) states that if we know the PDF at the input and output of the feature transformation $\mathbf{z} = T(\mathbf{x})$ under reference hypothesis H_0, written $p(\mathbf{x}|H_0)$ and $p(\mathbf{z}|H_0)$, then the function

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

is a PDF (it integrates to 1), and is a member of $\mathcal{P}(\mathbf{x})\{g(\mathbf{z})\}$. It follows that if we replace $g(\mathbf{z})$ with $p(\mathbf{z}_i|H_m)$, then we have a PDF estimate based on feature $T_i(\ )$:

$$\hat{p}(\mathbf{x}|H_m, T_i) = \frac{p(\mathbf{x}|H_0)}{p(\mathbf{z}_i|H_0)} \hat{p}(\mathbf{z}_i|H_m). \tag{2}$$

We can also create a fused estimate of $p(\mathbf{x})$ employing all L features by using a mixture PDF:

$$\hat{p}(\mathbf{x}|H_m) = \sum_i \alpha_i \left\{ \frac{p(\mathbf{x}|H_0)}{p(\mathbf{z}_i|H_0)} \hat{p}(\mathbf{z}_i|H_m) \right\}, \tag{3}$$

where $\sum_i \alpha_i = 1$. The fact that $p(\mathbf{x}|H_0)$ appears in (1) or (3) may seem to contradict our assumption at the outset that $\mathbf{x}$ might not be available. Note, however, that for many canonical reference hypotheses (exponential, Gaussian), the PDF $p(\mathbf{x}|H_0)$ depends on a scalar energy statistic. Thus, computing $p(\mathbf{x}|H_0)$ requires transmitting only a scalar such as $t_1(\mathbf{x}) = \sum_{n=1}^N x_n$, or $t_2(\mathbf{x}) = \sum_{n=1}^N x_n^2$.

B. Optimality: Why Maximum Entropy?

The reader may have already concluded that PDF projection lacks an optimality criterion. Although (1) is a member of $\mathcal{P}(\mathbf{x})\{g(\mathbf{z})\}$, is it the best one? The principle of maxumum entropy as a criterion for PDF design is well established [16]. The key idea is that the PDF estimate we choose should express not only out knowledge about the data, but also our ignorance. To illustrate this concept, we provide the following scenario.

In most real-world applications, the only "knowledge" we have about a PDF is through observations of some taining data. Suppose that we have a set of K training samples $\mathbf{x}_1,\ \mathbf{x}_2,\ \ldots,\mathbf{x}_K$. We have a number of proposed PDFs computed using (1) for various feature transformations $T_i(\mathbf{x})$. Let these projected PDFs be denoted by $p_i(\mathbf{x})$. We would like to determine which projected PDF (i.e which feature vector) provides a "better" fit to the data. We can compare the PDFs based on the average log-likelihood $L_i = \frac{1}{K}\sum_{n=1}^{K}\log p_i(\mathbf{x}_n)$, choosing the feature transformation that results in the largest value. But likelihood comparison by itself is misleading, because it only takes into account the training data (all of our knowledge), and does not take into account our ignorance. To meaasure our ignorance, we introduce another relevant quantity: the entropy of a distribution, $Q_i = -\int_{\mathbf{x}}\{\log p_i(\mathbf{x})\}\ p_i(\mathbf{x})\mathrm{d}\mathbf{x}$, which is the negative of the theoretical value of L_i, and is a generalization of the concept of variance. Distributions that spread the probability mass over a wider area have higher entropy since the average value of $\log p(\mathbf{x})$ is lower. The two concepts of Q and L are compared in Figure 2 in which we show three competing distributions: $p_A(\mathbf{x})$, $p_B(\mathbf{x})$, and $p_C(\mathbf{x})$. The vertical lines represent the location of the K training samples. If L_i is the average value of $\log p_i(\mathbf{x})$ at the training sample locations, then clearly $L_A \ll L_C \ll L_B$. But choosing $p_B(\mathbf{x})$ is very risky because it is over-adapted to the training samples. Clearly $p_B(\mathbf{x})$ has lower entropy since most of the probability mass is at places with higher likelihood. Therefore, it has achieved higher L at the cost of lower Q, a suspicious situation. On the other hand, $Q_A = Q_C$, but $L_C > L_A$. Therefore, $p_C(\mathbf{x})$ has achieved higher L than $p_A(\mathbf{x})$ without suffering lower Q, so choosing $p_C(\mathbf{x})$ over $p_A(\mathbf{x})$ is not risky. If we always choose among models that have maximum possible entropy for the given model parameterization, in this case determined by the choice of features, and are careful to separate data for training and testing, we likely to obtain better features and better generative models.

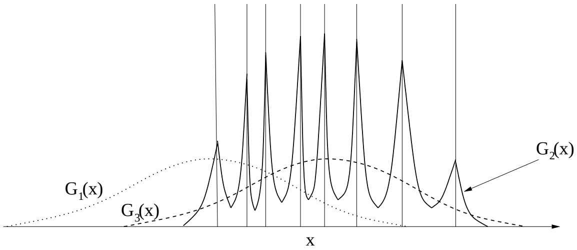

Fig. 2. Comparison of entropy Q and average log-likelihood L for three distributions. The vertical lines are the locations of training samples.

We therefore propose that when we combine or compare projected likelihood functions based on different features, always select the projected PDF with highest entropy. In the context of equation (1), that means the choice of H_0 should be the one that results in highest entropy.

C. Choosing H_0 for Maximum Entropy

It is surprisingly simple to choose H_0 in equation (1) for MaxEnt [14]. To best understand this choice from the standpoint of the generative model underlying PDF projection. To generate a sample of $G(\mathbf{x}; T, g, H_0)$ in (1), we first draw a sample $\mathbf{z}^*$ from the distribution $g(\mathbf{z})$. Next, we determine the manifold

$$\mathcal{M}(\mathbf{z}^*) = \{\mathbf{x} :\ T(\mathbf{x}) = \mathbf{z}^*\}. \tag{4}$$

The final step is to draw a sample $\mathbf{x}$ from this manifold, with distribution proportional to $p(\mathbf{x}|H_0)$. This means that the probability of drawing a sample $\mathbf{x}$ on the manifold is weighted proportional to the value of $p(\mathbf{x}|H_0)$. However, all our knowledge of $\mathbf{x}$ comes from $g(\mathbf{z})$, and as far as we know, all points on the manifold should be equally likely, and we should insist on this in order to express our ignorance. Therefore, if we can find a reference hypothesis $p(\mathbf{x}|H_0)$ such that no matter what the value of $\mathbf{z}^*$ is, that $p(\mathbf{x}|H_0)$ takes a constant value on any manifold $\mathcal{M}(\mathbf{z}^*)$, we have achieved our goal.

This reference hypothesis can be easily found as long as the feature $\mathbf{z}$ contains information about the size of $\mathbf{x}$ (expressed as a norm $\|\mathbf{x}\|$), what we call an *energy statistic* $t(\mathbf{x})$. Then, we may use a reference hypothesis of the exponential form

$$p(\mathbf{x}|H_0) = Ce^{-(t(\mathbf{x})/\alpha)^p},$$

which includes the canonical exponential and Gaussian distributions. Which we choose depends on the range of $\mathbf{x}$. If example of the elements of $\mathbf{x}$ are constrained to be positive, we use the energy statistic

$$t_1(\mathbf{x}) = \sum_{i=1}^{N} x_i, \tag{5}$$

and the exponential reference hypothesis

$$p(\mathbf{x}|H_0) = \prod_{i=1}^{N} e^{-x_i} = e^{-t_1(\mathbf{x})}. \tag{6}$$

Let $\mathbf{x}$ have support everywhere in $\mathcal{R}^N$. Then, we may use the energy statistic

$$t_2(\mathbf{x}) = \sum_{i=1}^{N} x_i^2 \tag{7}$$

and the Gaussian reference hypothesis

$$p(\mathbf{x}|H_0) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}} e^{-x_i^2/2} = (2\pi)^{-N/2}\ e^{-t_2(\mathbf{x})/2}, \tag{8}$$

If the elements of $\mathbf{x}$ are constrained to the unit interval $[0, 1]$, then the uniform distribution results in the MaxEnt PDF.

D. Generating samples of $G(\mathbf{x}; T, g, H_0)$

Clearly, (1) is a constructed PDF, so it is not of any canonical form for which sampling is defined. But, sampling from (1) is conceptually very simple when H_0 meets the criteria for MaxEnt using the two-step process [14]:

1) Draw a sample from $g(\mathbf{z})$, call it $\mathbf{z}^*$.
2) Draw a sample $\mathbf{x}$ from the manifold $\mathcal{M}(\mathbf{z}^*) = \{\mathbf{x} : T(\mathbf{x}) = \mathbf{z}^*\}$, with equal probabilty everywhere - that means draw from the uniform distribution on $\mathcal{M}(\mathbf{z}^*)$.

E. Implementation Issues

Despite the simplicity of (1), the implementation is not without difficulties. The primary difficulty in implementing (1) is the derivation of the feature PDF $p(\mathbf{z}|H_0)$. Although the use of an estimated PDF here seems reasonable, an estimated PDF cannot be accurate in the tails, where most data will fall - unless we use a reference hypothesis similar to real data - which precludes the use of a canonical reference hypothesis for MaxEnt. A breakthrough was made by Steven Kay who realized that for many features of interest in signal processing where no closed-form expression for $p(\mathbf{z}|H_0)$ may exist, the moment generating function can be derived and the saddle-point approximation (SPA) can be used. A publication appeared in 2000 applying the SPA to linear functions of exponential random variables, what underlies a large number of widely-used spectral features [17]. Nuttall applied the SPA do derive the distribution of quadratic functions of correlated random variables [18] and order statistics [19]. The SPA, together with the chain-rule (See next section) can be used to analyze virtually any feature of practical use[1].

F. 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 to impossible to carry out the necessary derivation of $p(\mathbf{z}|H0)$, 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),

$$G(\mathbf{x};g,H_0)= \left[\frac{p(\mathbf{x}|H_{0x})}{p(\mathbf{y}|H_{0x})}\right] \left[\frac{p(\mathbf{y}|H_{0y})}{p(\mathbf{w}|H_{0y})}\right] \left[\frac{p(\mathbf{w}|H_{0w})}{p(\mathbf{z}|H_{0w})}\right] g(\mathbf{z}), \quad (9)$$

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. 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, (9) 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-C hold individually at each stage, then it is the maximum entropy member. When drawing samples from $G(\mathbf{x}; g, H_0)$, 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.

G. Noteworthy Publications

The predecessor of PPT was the class-specific features method that was based on sufficient statistics [20], [21], [22], [23]. Although the MaxEnt extension of PDF projection was not realized until recently, the conditions for MaxEnt, specifically the use of the energy statistic and the canonical Gaussian and exponential PDFs, were adhered to in past applications on the basis of scale invariance ([6], Section II.B). Papers exploring PPT and its applications include [24], [6], [25], [26], [27], [28]. The multi-resolution HMM [29] applies the PPT to segment data on-the-fly, allowing feature extraction of varying time and frequency resolution to be joined in a single rigorous statistical model.

H. Summary of Theoretical Results

We have described a new theoretical framework for statistical inference using a generative method based on projected PDFs. Let there be a high-dimensional data sample $\mathbf{x} \in \mathcal{R}^N$ or $\mathbf{x} \in \mathcal{P}^N$, that we potentially cannot directly observe. Instead, we have measured L possibly multi-dimensional features $\mathbf{z}_i = T_i(\mathbf{x})$, $1 \leq i \leq L$ and have previously estimated the corresponding feature PDFs $\hat{p}(\mathbf{z}_i|H_m)$ under each of M hypotheses H_m. Then, we may form an estimate of $p(\mathbf{x}|H_m)$ using a single feature $\mathbf{z}_i = T_i(\mathbf{x})$

$$\hat{p}(\mathbf{x}|H_m, T_i) = \frac{p(\mathbf{x}|H_0)}{p(\mathbf{z}_i|H_0)} p(\mathbf{z}_i|H_m),$$

or can employ all L features by using a mixture PDF:

$$\hat{p}(\mathbf{x}|H_m) = \sum_i \alpha_i \left\{ \frac{p(\mathbf{x}|H_0)}{p(\mathbf{z}_i|H_0)} p(\mathbf{z}_i|H_m) \right\},$$

where $\sum_i \alpha_i = 1$. This may be calculated without observing $\mathbf{x}$ since the information necessary to compute $p(\mathbf{x}|H_0)$ is contained in $\mathbf{z}_i$.

IV. Classification Experiment

One important application of PDF projection is to employ multiple feature vectors in classifiers without incuring the full dimensionality of the union of the features. We present now a vivid demonstration of this ability.

A. Simulated Data

Let the data $\mathbf{x} \in \mathcal{R}$ be generated accoring to one of five synthetic data classes, examples of which are shown in Figure 3:

1) Class 1: Independent identically-distributed zero-mean Gaussian noise of variance 1, denoted by $\mathcal{N}(0, 1)$ noise.
2) Class 2: Sinewaves. To $\mathcal{N}(0, 1)$ noise, we add a sinusoid with random phase and amplitude 1.5 and with frequency centered at 0.1666 relative frequency (0.5 = Nyquist freq) and with a random Gaussian offset of standard deviation 0.005.
3) Class 3: Auto-regressive noise with all-pole filter coeffients $\mathbf{a} = [1, -.9375, 0.8789]$ and innovation variance 1. When generating data, N initial samples are discarded. This is added $\mathcal{N}(0, 1)$ noise.
4) Class 4: Impulsive noise created from the 6-th power of Gaussian samples and scaled by .0125. This is added to $\mathcal{N}(0, 1)$ noise.

[1] Note that a feature needs to be analyzable in the sense that $p(\mathbf{z}|H_0)$ can be derived. If a feature cannot be analyzed, there often is an analyzable replacement.

5) Class 5: Modulated noise. A vector of N samples of $\mathcal{N}(0,1)$ noise is divided into four equal sections which are scaled by the factors [2, 0.5, 2, 0.5]. This is added to $\mathcal{N}(0,1)$ noise.

After generating the data, a random scale factor was then applied equal to the square-root of an exponentially-distributed RV.

B. Features

We defined 5 feature sets, more or less corresponding to the five synthetic data classes. Each feature transformation was analyzed using $\mathcal{N}(0,1)$ as the reference hypothesis H_0.

1) Power. The log of the power statistic $t_2(\mathbf{x})$ defined at the end of Section III-A. Under H_0, this feature is a chi-square statistic of N degrees of freedom.
2) Spectral order-statistics. Take the FFT of $\mathbf{x}$, then compute the square-magnitide of the bins, sort them and keep the K largest bins. The features are (a) The K frequency indexes of the K largest bins, (b) the squared magnitudes, (c) the remaining energy in $\mathbf{x}$. We used $K = 2$. To analyze these features, we applied the SPA [19]. Feature dimension: $2K+1 = 5$.
3) Auto-regressive features (reflection coeffs) of order $P = 3$. The analysis of these features is covered in ([14], Section V.B, page 2821). Feature dimension: $P + 1 = 4$.
4) Amplitude order-statistics. We squared the data samples, then sorted them, keeping the K top values. These K values, and the sum of the remaining energy were retained as features. To analyze these features, we applied the SPA [19]. We used $K = 5$. Feature dimension: $K + 1 = 6$.
5) DCT of instantaneous power. We squared the samples, then took the first 6 coefficients of the DCT. This feature is an application of the linear function of chi-square random variables. The SPA for the distribution of the feature under H_0 has been described in [17]. Feature dimension: 6.

The total feature dimension (if all features are grouped together) is 22.

C. Experimental Setup

We conducted a series of experiments as a function of the number of training samples, with three trials each. In each trial, we generated n training samples from each data class, trained the classifiers, then tested using 1000 independent samples of each class. There were therefore 5000 test samples in each trial, 15,000 testing samples for each value of n. We used $n = [5, 7, 10, 15, 25, 50, 100, 200, 500, 1000, 2000, 5000, 10000]$.

D. Classifiers

Let H_i, $1 \leq i \leq M$ be the class hypotheses ($M = 5$). We tested the following classifiers:

1) Class-dependent PDF projection (PPT). We applied (2) as a straight Neyman-Pearson classifier:

$$\arg \max_{i=1}^{M} \hat{p}(\mathbf{x}|H_m, T_i),$$

where to test each class hypothesis H_i, we used the projected PDF created using the corresponding feature transformation $T_i(\mathbf{x})$ from Section IV-B. Each feature PDF $\hat{p}(\mathbf{z}_i|H_m)$ is obtained by a Gaussian mixture model (GMM) trained on the n training samples of class H_i. The numerator term in (2) , $p(\mathbf{x}|H_0)$ is just the theoretical PDF of $\mathbf{x}$ for independent $\mathcal{N}(0,1)$ noise. The denominator PDF $p(\mathbf{z}_i|H_0)$ is obtained separately for each feature transformation as described in Section IV-B.
2) Gaussian mixture model (GMM). We gathered all 22 features together into a feature super-vector which was used to train a GMM with mixture components.
3) Multi-layer perceptron (MLP). The 22-dimensional feature super-vector was used to train a multi-layer perceptron neural network with an output layer of 5 nodes, and a hidden layer of 10 nodes, using back-propagation [33].
4) Support vector machine (SVM). The 22-dimensional feature super-vector was used to train a SVM. We used the SVM-lite toolkit [34] with linear kernel (radial basis function kernel gave similar results).

E. Results

Experimental results are shown in Figure 4 as a function of number of training samples per class n. Each curve is composed of three lines: minimum, maximum and mean of the three trials. The PPT classifier significantly outperformed all the other classifiers, especially at low training samples, a result of the lower feature dimension for PPT. The MLP out-performed the GMM, but only at high n. Various arrangements of the MLP were tried, for example, with no hidden layer, but the arrangement with 10 hidden nodes worked best. Interestingly, both the GMM and the MLP out-performed the SVM. Several SVM kernel functions were tried, with little difference noted. Most important to note is that the PPT, which had feature dimension no greater than 6, reached best performance at lower training samples than the other methods, which used a 22-dimensional feature vector.

V. Applications

The method of MEPP solves a fundamental problem in statistical inference and has many applications. The range of potential applications for maximum entropy PDF projection is wide and we have only scratched the surface.

1) Feature selection and class-dependent features. Choosing features using the method of Section III-B has been successfully demonstrated [31]. But, the practitioner is free to use one or more feature vectors, and may even use a different feature vector to test each hypothesis (2). We demonstrated this capability in Section IV.
2) Feature mixtures. It is also possible to form mixture densities using several feature vectors (3). Improved classification performance has been demonstrated with this technique [30], [31].
3) Compressive sensing. When data must be compressed, and later inference about the original data must be made, $G(\mathbf{x}; T, g, H_0)$ can be used as an optimal PDF estimate. When the same original data

is observed through multiple sensors, the information can be fused statistically.

4) Multi-resolution processing. If the segmentation of the data is now known *a priori*, then various segmentations can be tested using PDF projection. This way, a common statisical model can consider processing at various resolutions. This has been call multi-resolution HMM [32], [29].
5) Monte Carlo methods. The raw data PDF (1) is a generative model from which we may generate samples of $\mathbf{x}$. This opens new research directions in applying Monte Carlo methods, which rely on generating samples of a model density, to high-dimensional data.
6) Neural Networks. There is much current interest in generative methods, as explained in the Introduction. Generative methods are now used in conjunction with neural networks. Since a Neural Network can be seen as a feature extraction, it can be seen as a generative model using maximum entropy PDF projection. Therefore, we see the potential to create new types of generative models using maximum entropy PDF projection and use them in close cooperation with conventional neural networks.

VI. Conclusions

In this paper, we have introduced the concept of fusing information from multiple features into a common decision rule using maximum entropy PDF projection. We demonstrated the concept by classifying synthetic data using a separate features to test each class hypothesis, achieving superior performance to existing methods.

References

[1] J. Neyman and E. S. Pearson, "On the problem of the most efficient tests of statistical hypotheses," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 231*, p. 289337, 1933.

[2] J. Berger, *Statistical Decision Theory and Bayesian Analysis*. Berlin Heidelberg: Springer, 1993.

[3] V. Vapnik, *The Nature of Statistical Learning*. Springer, 1999.

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

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

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

[7] R. E. Bellman, *Adaptive Control Processes*. Priceton, New Jersey, USA: Princeton Univ. Press, 1961.

[8] K. Fukunaga, *Introduction to Statistical Pattern Recognition (2nd ed)*. San Diego: Academic Press, 1990.

[9] Duda and Hart, *Pattern Classification and Scene Analysis*. Wiley, 1973.

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

[11] A. Y. Khincin, *Mathematical Foundations of Information Theory*. Mineaola, NY: Dover, 1957.

[12] T. Cover and J. Thomas, *Elements of Information Theorey*. John Wiley and Sons, 1991.

[13] S. Basu, C. A. Micchelli, and P. Olsen, "Maximum entropy and maximum likelihood criteria for feature selection from multivariate data," in *Proceedings the 2000 IEEE international symposium on Circuits and Systems, ISCAS 2000, vol 3*, Geneva, May 2000, pp. 267–270.

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

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

[16] E. T. Jaynes, "On the rationale of maximum-entropy methods," *Proceedings of IEEE*, vol. 70, no. 9, pp. 939–952, 1982.

[17] S. M. Kay, A. H. Nuttall, and P. M. Baggenstoss, "Multidimensional probability density function approximation for detection, classification and model order selection," *IEEE Trans. Signal Processing*, pp. 2240–2252, Oct 2001.

[18] A. H. Nuttall, "Saddlepoint approximation and first-order correction term to the joint probability density function of M quadratic and linear forms in K Gaussian random variables with arbitrary means and covariances," *NUWC Technical Report 11262*, December 2000.

[19] ——, "Joint probability density function of selected order statistics and the sum of the remaining random variables," *NUWC Technical Report 11345*, January 2002.

[20] P. M. Baggenstoss, "Class-specific features in classification." *IEEE Trans Signal Processing*, pp. 3428–3432, December 1999.

[21] S. Kay, "Sufficiency, classification, and the class-specific feature theorem," *IEEE Trans. Information Theory*, vol. 46, no. 4, pp. 1654–1658, July 2000.

[22] H. C. B. Caputo, "A Marxist approach to object recognition: kernel-specific classifiers," in *Proceedings of the 2004 symposium on image analysis (SSBA)*, 2004.

[23] Z. J. Wang and P. Willett, "Joint segmentation and classification of time-series using class-specific features," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 2, pp. 1056–1067, Apr 2004.

[24] P. M. Baggenstoss, "The class-specific classifier: Avoiding the curse of dimensionality (tutorial)," *IEEE Aerospace and Electronic Systems Magazine, special Tutorial addendum*, vol. 19, no. 1, pp. 37–52, January 2004.

[25] V. Estellers and P. M. Baggenstoss, "Class-specific classifiers in audio-visual speech recognition," in *EUSIPCO 2010*, 2010.

[26] Y. Sun and P. Willett, "Automated classification of signals with duration-dependent segments via class-specific features and gibbs sampling," in *IEEE Aerospace Conference*, 2012, pp. 1–11.

[27] T. Beierholm and P. M. Baggenstoss, "Speech music discrimination using class-specific features," in *Proc. ICPR 2004*, 2004.

[28] B. Tang, H. He, P. Baggenstoss, and S. Kay, "A bayesian classification approach using class-specific features for text categorization," *IEEE Transactions on Knowledge and Data Engineering*, p. (Accepted Feb 2016), 2016.

[29] P. M. Baggenstoss, "A multi-resolution hidden markov model using class-specific features," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5165–5177, Oct 2010.

[30] ——, "Class-specific model mixtures for the classification of time-series," 2014.

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

[32] ——, "A multi-resolution hidden markov model using class-specific features," *Proceedings of EUSIPCO 2008, Lausanne, Switzerland*, Aug 2008.

[33] V. Vapnik, *Statistical Learning Theory*. John Wiley, 1998.

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

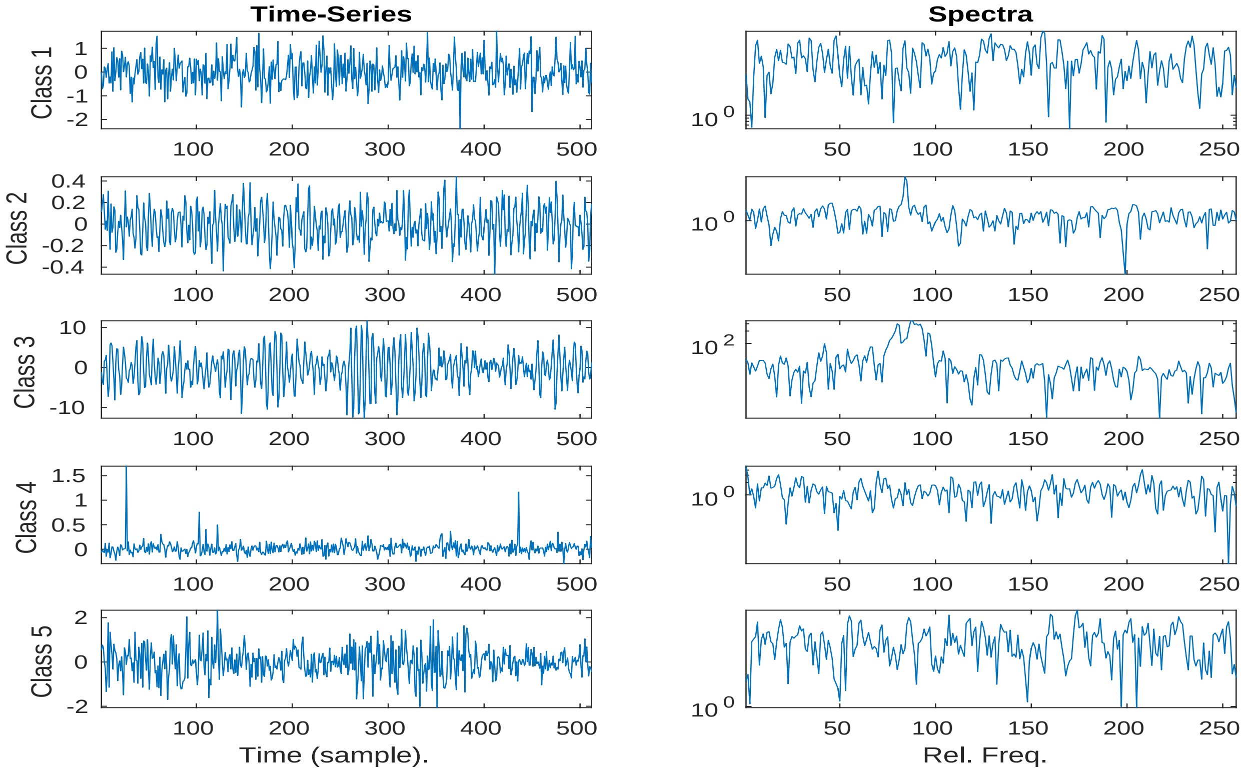

Fig. 3. Typical samples of each synthetic data classes, time-series (left) and spectrum (right).

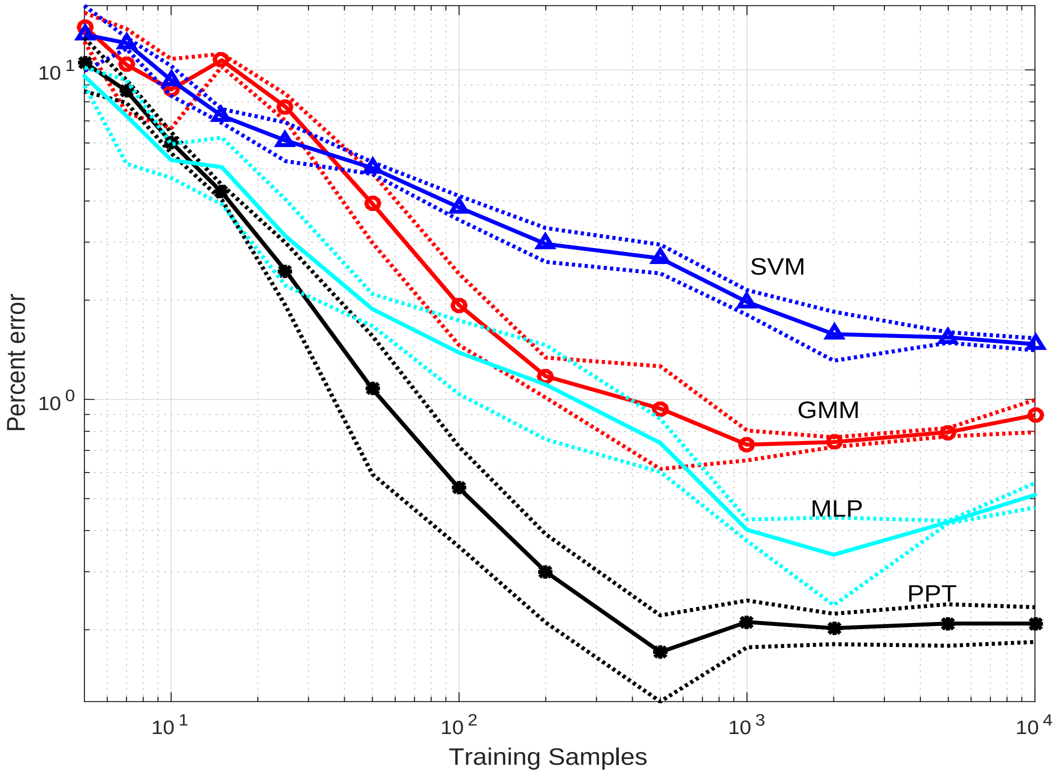

Fig. 4. Classifier performance as a function of number of training samples for four classifiers.