Uniform Manifold Sampling (UMS): Sampling the Maximum Entropy PDF

Dr. Paul M. Baggenstoss [1]

***Abstract*—Maximum entropy PDF projection (MEPP) is a way to construct generative models from feature transformations. Corresponding to each dimension-reducing feature mapping, such as a feed-forward neural network or an algorithm to calculate linear-prediction coefficients from time-series, and given a prior distribution for the features, MEPP finds a unique generative model for the input data, which subject to mild requirements, is maximum entropy (MaxEnt) among all probability density functions (PDFs) that are consistent with the given feature prior. In this paper, we consider the problem of sampling from these MaxEnt projected PDFs. The sampling process consists of drawing a sample from the given feature prior distribution, then drawing samples uniformly distributed on the inversion set (set of input samples consistent with the drawn feature value, usually a manifold). The process is called uniform manifold sampling (UMS). We describe UMS for simple non-linear and iterative feature transformations, then focus on linear transformations with input data constraints ($x_i > 0$ or $0 \leq x_i \leq 1$), which require MCMC-based sampling. We discover that the manifold centroid (sample mean for a fixed feature value) is useful as a deterministic MaxEnt feature inversion solution. We show how to predict the centroid efficiently without sampling and demonstrate its usefulness in speeding up MCMC by an order of magnitude, and in spectral estimation and image reconstruction. Finally, we provide an example of UMS in a classification experiment in which we use Monte Carlo integration to create true generative models from arbitrary classifiers.**

I. Introduction

A. Background

Due to their direct solution of the underlying inference problem, discriminative methods have dominated classification methods for many years [1]. Despite this, generative methods have advantages - they can generalize better to un-forseen changes in data make-up, are modular (by class), can make use of unlabeled data, and can be easily interrogated - to see what they have learned about a given class. In fact, generative methods are now seeing a re-birth, for example, in a form of Bayesian belief network called deep belief network (DBN) [2]. These new generative models rival or exceed the performance of discriminative models [2].

B. PDF Projection

The method of PDF projection [3], and more recently maximum entropy (MaxEnt) PDF projection [4], can be seen as belonging to this re-birth of generative methods. But, PDF projection is distinct because it is an *implied* generative model with non-explicit sampling. It is also completely general and has no favored structure. All that is required for PDF projection is a known feature mapping $\mathbf{z} = T(\mathbf{x})$, where T is some fixed dimension-reducing transformation satisfying mild regularity conditions. For example, T could be a feed-forward neural network. The feature transformation and the implied generative model are duals. A special case of this duality relationship can be seen in the inference filters that are used to estimate the hidden variables in a Bayesian belief network [2]. MaxEnt PDF projection formalizes this duality relationship using the maximum entropy principle. But, because MEPP constructs an *implied* generative model, sampling is not explicit.

[1]Fraunhofer FKIE, Wachtberg, Germany, p.m.baggenstoss@ieee.org

C. Why Sample?

A generative model is a mathematical description of the data generation process. Mostly, however, generative models are used for inference - to test the likelihood that a given input data sample was generated by the model. There are, however, compelling reasons to sample:

1) Qualitative validation of the distribution $p(\mathbf{x})$, by observing the quality and suitability of the generated samples [2].
2) Monte Carlo integration (MCI). Consider an arbitrary function $h(\mathbf{x})$ with support $\mathcal{X}$. Let $p_p(\mathbf{x})$ be a known *proposal distribution*, having the same support $\mathcal{X}$, from which samples may be drawn. We approximate the integral of $h(\mathbf{x})$ over $\mathcal{X}$ by

$$C = \int_{\mathbf{x}\in\mathcal{X}} h(\mathbf{x})\mathrm{d}\mathbf{x} \simeq \frac{1}{K}\sum_{i=1}^{K} \frac{h(\mathbf{x}_i)}{p_p(\mathbf{x}_i)}, \qquad (1)$$

   where the samples x_i are drawn from $p_p(\mathbf{x})$. In order to promote efficient MCI, $h(\mathbf{x})$ and $p_p(\mathbf{x})$ must be compatible - so that $p_p(\mathbf{x})$ "covers" $h(\mathbf{x})$ well or that high-likelihood regions of $h(\mathbf{x})$ cannot lie in extremely low-likelihood regions of $p_p(\mathbf{x})$. We provide an example in Section VI.
3) Non-linear mixtures of generative models. Let $h(\mathbf{x}) = \left(\sum_i w_i p_i(\mathbf{x})^{1/\alpha}\right)^{\alpha}$. The parameter α is related to the "temperature" parameter of the "softmax" function that is widely used in machine learning and tends to control the "hardness" (or "softness") of mixing. The linear mixture $p_p(\mathbf{x}) = \sum_i w_i p_i(\mathbf{x})$ is a compatible proposal distribution. Let C be the integral of $h(\mathbf{x})$ obtained by MCI. Then, $h(\mathbf{x})/C$ is a valid generative model that can be sampled (using rejection sampling [5]) and used to estimate factor α by maximum likelihood (ML). We provide an example in Section VI.

4) Hybrid generative/discriminative models. Let $y_m(\mathbf{x})$ be a discriminative classifier output that approximates the posterior class probability $y_m(\mathbf{x}) \simeq p(H_m|\mathbf{x})$. Now, given an arbitrary generative model $p(\mathbf{x}|H_m)$, we form the product

$$h_m(\mathbf{x}) = p(\mathbf{x}|H_m)\, e^{(y_m(\mathbf{x})-1)/a}. \tag{2}$$

Parameter a controls the relative effect of $y_m(\mathbf{x})$. The PDF $p(\mathbf{x}|H_m)$ is a compatible proposal distribution for MCI. Let C_m be the integral of $h_m(\mathbf{x})$ obtained by MCI. Then, $h_m(\mathbf{x})/C_m$ is a valid generative model that can be sampled and used to estimate a by ML. We provide an example in Section VI.

5) When $p(\mathbf{x})$ is a projected PDF, sampling can be used to better understand feature inversion methods, as we will explain below when we discuss the manifold centroid.

D. Paper outline and Main Contributions

In section II, we lay the theoretical foundation for the paper by presenting the main theorems underlying PDF projection, MEPP and UMS. In Section III, we describe UMS for some simple feature transformations and for an iterative (maximum likelihood) estimator. In Section IV, we discuss UMS for linear transformation when $x_i > 0$, which requires MCMC sampling. We develop a deterministic way to estimate the manifold centroid, which is then used to speed up MCMC and serves as a feature inversion method corresponding to classical MaxEnt methods. In Section V, we consider when $0 \leq x_i \leq 1$, adapting the MCMC approach, and show that the manifold centroid corresponds to a new entropy measure based on the truncated exponential distribution. We demonstrate it in reconstructing images. Finally, in Section VI, we provide a classification problem illustrating the sampling motivations spelled out above.

II. Mathematical results

A. PDF Projection

Let there be a feature transformation or *mapping*

$$T : \mathcal{X} \to \mathcal{Z}, \quad \mathcal{X} \subset \mathcal{R}^N, \quad \mathcal{Z} \subset \mathcal{R}^D, \quad D < N. \tag{3}$$

We write this simply as $\mathbf{z} = T(\mathbf{x})$. We assume that T is "onto" so that all members of $\mathcal{Z}$ are mapped from at least one member of $\mathcal{X}$. There are some mild regularity conditions assumed for T which are mentioned when discussing Theorem 1. Given some specified feature PDF $g(\mathbf{z})$ on $\mathcal{Z}$, there exists a set of PDFs on $\mathcal{X}$, denoted by $\{\mathcal{P} : T, g\}$ $G(\mathbf{x})$, that are consistent with $g(\mathbf{z})$, meaning that if $G \in \{\mathcal{P} : T, g\}$, then samples drawn from G and passed through T will have exactly distribution g. PDF projection [3] is a means of constructing a member of $\{\mathcal{P} : T, g\}$ based on a reference hypothesis H_0. The constructed PDF is called the *projected* PDF.

Theorem 1: **(PDF Projection theorem)**, *see [6], [4]. Consider the feature mapping (3). Let $g(\mathbf{z})$ be an arbitrary feature PDF with support $\mathcal{Z}$. Let $p(\mathbf{x}|H_0)$ be a reference distribution with support $\mathcal{X}$. Let $p(\mathbf{z}|H_0;T)$, the distribution imposed on $\mathcal{Z}$ when $\mathbf{z} = T(\mathbf{x})$ and $\mathbf{x} \sim p(\mathbf{x}|H_0)$. Let $p(\mathbf{z}|H_0;T)$ be non-zero and have finite value everywhere on $\mathcal{Z}$. Then, the function*

$$G(\mathbf{x}; H_0, T, g) = \frac{p(\mathbf{x}|H_0)}{p(\mathbf{z}|H_0;T)} g(\mathbf{z}), \quad \mathbf{z} = T(\mathbf{x}) \tag{4}$$

is a PDF (integrates to 1 over $\mathcal{X}$), and is a member of $\{\mathcal{P} : T, g\}$.

For a proof, see [6] or [4], Theorem 2. Note that H_0 is a mathematical concept, and does not need to represent any type of "noise-only" condition or realistic data or such. To see how assuming that $p(\mathbf{z}|H_0;T)$ exists and has finite value on $\mathcal{Z}$ imposes certain conditions on $T(\mathbf{x})$, we write

$$p(\mathbf{z}|H_0;T) = \int_{\mathbf{x}\in\mathcal{M}(z;T)} p(\mathbf{x}|H_0)\, \mathrm{d}\mathbf{x}, \tag{5}$$

where the integral is carried out on the level set

$$\mathcal{M}(\mathbf{z};T) = \{\mathbf{x} : \ T(\mathbf{x}) = \mathbf{z}, \ \ \mathbf{x} \in \mathcal{X}\}. \tag{6}$$

The existence of (5) implies that $p(\mathbf{x}|H_0)$ is (Lebesgue) integrable on any level set $\mathcal{M}(\mathbf{z};T)$, an implied regularity condition for T. In the following, we refer to $\mathcal{M}(\mathbf{z};T)$ as "manifold", which implies that the level set is locally euclidean [7], which further implies T is smooth. Level sets of smooth functions (conventional level sets [8]) are manifolds [7]. The requirements for the existence of integral (5) do not require manifolds, but the level sets we encounter in this paper all produce at least closed subsets of manifolds.

The generality of (4) is strengthened by the following theorem.

Theorem 2: **(Completeness of PDF Projection)**. *Any member of $\{\mathcal{P} : T, g\}$ can be constructed using (4).*

This theorem is credited to Steven Kay [9]. Proof: See [4] Section II.A and Theorem 1. The importance of the theorem is that we can seek the maximum entropy member of $\{\mathcal{P} : T, g\}$ just by choosing H_0.

The following corrolary describes how to draw samples from $G(\mathbf{x}; H_0, T, g)$.

Corrolary 1: **(Sampling from projected PDF).** *To draw samples from $G(\mathbf{x}; H_0, T, g)$, we first draw a sample $\mathbf{z}^*$ from $g(\mathbf{z})$, then draw a sample $\mathbf{x}$ from the manifold $\mathcal{M}(\mathbf{z}^*;T)$ with a probability distribution on the manifold proportional to $p(\mathbf{x}|H_0)$.*

The above corrolary follows as a result of showing the completeness property (See [4] Section II.A and Theorem 1).

We call the second part of the sampling process described in Corrolary 1 *manifold sampling*. It consists of drawing a sample $\mathbf{x}$ on the manifold with distribution proportional to $p(\mathbf{x}|H_0)$. We may view this "manifold distribution", a function that integrates to 1 on the manifold, as

$$\mu(\mathbf{x}|\mathbf{z}^*;T,H_0) = \frac{p(\mathbf{x}|H_0)\ \delta(\mathbf{z}^*,T(\mathbf{x}))}{\int_{\mathbf{x}\in\mathcal{M}(\mathbf{z}^*;T)}\ p(\mathbf{x}|H_0)\ \mathrm{d}\mathbf{x}}, \tag{7}$$

where $\delta(\mathbf{z}^*,T(\mathbf{x})) = 1$ if $\mathbf{z}^* = T(\mathbf{x})$ and zero otherwise. By its definition, $\mu(\mathbf{x}|\mathbf{z}^*;T,H_0)$ integrates to 1 on the manifold and can be interpreted as a posterior $p(\mathbf{x}|\mathbf{z})$, which makes clear the role of $p(\mathbf{x}|H_0)$: it shapes the manifold distribution. But, because $\mathbf{z} = T(\mathbf{x})$ is a deterministic mapping, $p(\mathbf{x}|\mathbf{z})$ is not a proper distribution - all of its probability mass lies on a manifold of zero volume.

B. Why Maximum Entropy?

The principle of maximum entropy is a well established criterion for PDF design [10]. We now explain why we should optimize (4) to maximize the entropy over H_0 - and thereby take into consideration not only our knowledge about the data, but also our ignorance.

Often, the only "knowledge" we have about a PDF is through observations of some data. Consider a set of K training samples $\mathbf{x}_1,\ \mathbf{x}_2,\ \ldots, \mathbf{x}_K$ and a number of proposed PDFs computed using (4) for various feature transformations $T_l(\mathbf{x})$, denoted by $G_l(\mathbf{x})$. If we select the PDF based on maximizing the average projected log-likelihood $L_l = \frac{1}{K}\sum_{n=1}^{K} \log G_l(\mathbf{x}_n)$, the result will be misleading because it only takes into account our knowledge (data), but not our ignorance. We measure our ignorance using the entropy $Q_l = -\int_{\mathbf{x}} \{\log G_l(\mathbf{x})\}\ G_l(\mathbf{x}) \mathrm{d}\mathbf{x}$, which is the negative of the theoretical value of L_l. If the probability mass is spread over a wider area, the average value of $\log p(\mathbf{x})$ is lower, so Q_l is higher. The two concepts of Q and L are illustrated in Figure

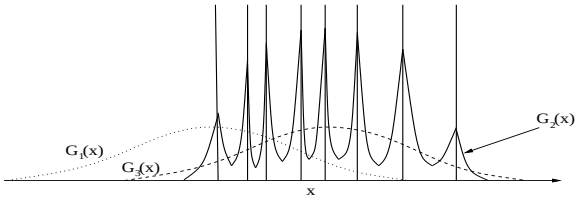

Fig. 1. Comparison of entropy Q and average log-likelihood L for three distributions. The vertical lines are the locations of training samples.

1 in which we show three competing distributions: $G_1(\mathbf{x})$, $G_2(\mathbf{x})$, and $G_3(\mathbf{x})$. The vertical lines represent the location of the K training samples. If L_l is the average value of $\log G_l(\mathbf{x})$ at the training sample locations, then clearly $L_1 \ll L_3 \ll L_2$. But choosing $G_2(\mathbf{x})$ is very risky because it is over-adapted to the training samples and 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, $Q_1 = Q_3$, but $L_3 > L_1$. Therefore, $G_3(\mathbf{x})$ has achieved higher L than $G_1(\mathbf{x})$ without suffering lower Q, so choosing $G_3(\mathbf{x})$ over $G_1(\mathbf{x})$ is not risky. We therefore propose that when we compare projected likelihood functions based on different features, we should select the projected PDF with highest entropy for the given feature transformation T. In the context of equation (4), that means the choice of H_0 should be the one that results in highest entropy.

C. MaxEnt PDF Projection

Maximum entropy PDF projection [4] is a means of finding the unique member of $\{\mathcal{P} : T, g\}$ with highest entropy, more precisely: to find the H_0 that produces $G(\mathbf{x}; H_0, T, g)$ with highest entropy. The entropy of $G(\mathbf{x}; H_0, T, g)$ is given by

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

It can be shown (See [4], equation 8) that this can be expanded as follows:

$$Q_G = Q_g + \int_{\mathbf{z}} Q_{\mu|z;H_0}\ g(\mathbf{z})\ \mathrm{d}\mathbf{z} \tag{8}$$

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

$$Q_{\mu|z;H_0} = -\int_{\mathbf{x}\in\mathcal{M}(z;T)} \log \mu(\mathbf{x}|\mathbf{z};T,H_0)\ \mu(\mathbf{x}|\mathbf{z};T,H_0)\mathrm{d}\mathbf{x}, \tag{9}$$

where $\mu(\mathbf{x}|\mathbf{z};T,H_0)$ comes from (7). Since Q_g is fixed, to absolutely maximize $Q_{\mu|z;H_0}$ (i.e. for each $g(\mathbf{z})$), $Q_{\mu|z;H_0}$ must be maximized for *each* $\mathbf{z}$. This is achieved if $\mu(\mathbf{x}|\mathbf{z};T,H_0)$ is the uniform distribution, which is the MaxEnt distribution on regions of compact support. But, in (7), $\mu(\mathbf{x}|\mathbf{z};T,H_0)$ is shaped by $p(\mathbf{x}|H_0)$. Therefore, we have two requirements, (a) $p(\mathbf{x}|H_0)$ must be of constant value on any manifold, and (b) all manifolds $\mathcal{M}(\mathbf{z};T)$ must be compact. There are two ways to achieve these requirements depending on $\mathcal{X}$.

When $\mathcal{X}$ is itself a compact set, we can make $p(\mathbf{x}|H_0)$ the uniform distribution. Then, so long as $\mathcal{M}(\mathbf{z};T)$ is compact for all $\mathbf{z}$ [1], then $\mu(\mathbf{x}|\mathbf{z};T,H_0)$ will be a proper uniform distribution for all $\mathbf{z}$, which has maximum entropy. Alternatively, when $\mathcal{X}$ is infinite in extent, the manifold can be forced to be compact by the inclusion of an *energy statistic* in $\mathbf{z}$ (first proposed in [4]). The solution for compact $\mathcal{X}$ and the solution for unbounded $\mathcal{X}$ are formalized by the following two theorems.

Theorem 3: **Maximum Entropy PDF Projection - Compact $\mathcal{X}$.** *Starting with the same assumptions as Theorem 1, we further assume that $\mathcal{X}$ is a compact set and $\int_{\mathbf{x}\in\mathcal{X}} \mathrm{d}\mathbf{x} = a < \infty$. Furthermore, we assume that $\mathcal{M}(\mathbf{z};T)$ is a compact set for all $\mathbf{z} \in \mathcal{Z}$. Then, the PDF*

$$G^*(\mathbf{x};T,g) = \frac{a^{-1}}{p(\mathbf{z}|H_0;T)} g(\mathbf{z}), \tag{10}$$

where $p(\mathbf{z}|H_0;T)$ is the distribution of $\mathbf{z}$ under the uniform assumption $p(\mathbf{x}|H_0) = a^{-1}$, is the member of $\{\mathcal{P} : T, g\}$ with highest entropy.

Proof. By Theorem 2, we can seek the maximum entropy PDF within the context of PDF projection - by selecting H_0 to maximize the second term in (8) over H_0. But, this is the expected value of the manifold entropy with expectation taken over $g(\mathbf{z})$. Under the assumptions of the theorem, the uniform distribution on $\mathcal{M}(\mathbf{z};T)$ maximizes (9) for any value of $\mathbf{z}$, thus globally maximizing (8).

The second case considers when $\mathcal{X}$ is not compact.

Theorem 4: **Maximum Entropy PDF Projection - Unbounded $\mathcal{X}$.** *Starting with the same assumptions as Theorem 1, we further assume that there exists a function f such that*

$$f(\mathbf{z}) = f(T(\mathbf{x})) = \|\mathbf{x}\| \tag{11}$$

for some norm $\|\mathbf{x}\|$ valid in $\mathcal{X}$. We further assume that for all finite $\mathbf{z} \in \mathcal{Z}$, $\mathcal{M}(\mathbf{z};T)$ is a compact set. Then, if the reference distribution can be written in the form

$$p(\mathbf{x}|H_0) = h(T(\mathbf{x})) \tag{12}$$

[1] Just because $\mathcal{X}$ is compact does not automatically imply $\mathcal{M}(\mathbf{z};T)$ is compact even though $\mathcal{M}(\mathbf{z};T) \in \mathcal{X}$.

for some function h, then the projected PDF (4) is the member of $\{\mathcal{P}:T,g\}$ with highest entropy.

The proof is provided in [4], but we provide an outline. Clearly (12) must be constant on the manifold (6) since $T(\mathbf{x})$ is fixed. Therefore, the manifold distribution (7) is the uniform distribution. Also, by (11), since $\mathbf{z}$ is fixed, $\|\mathbf{x}\|$ is constrained to a constant on the manifold, so the manifold itself must be bounded. Since the manifold is bounded, the further assumption that the manifold is a compact set is not very restrictive. The uniform distribution is the MaxEnt distribution on the compact manifold [11].

The most straight-forward way to achieve both (11) and (12) simultaneously is to choose a scalar statistic $t(\mathbf{x})$ that we call *energy statistic* (ES). We assume that $\|\mathbf{x}\|$ can be computed from $t(\mathbf{x})$ and that $t(\mathbf{x})$ can be computed from $T(\mathbf{x})$. We can then choose a reference distribution in the exponential family

$$p(\mathbf{x}|H_0) = C\exp\left\{-|t(\mathbf{x})/a|^p\right\}, \quad p \geq 1.$$

This family includes standard normal and exponential distributions.

Example 1: Let $\mathbf{x}$ have support in $\mathcal{P}^N$, defined as the positive quadrant of $\mathcal{R}^N$, where $x_i > 0,\ \forall i$. The statistic

$$t_1(\mathbf{x}) = \sum_{i=1}^{N} x_i \tag{13}$$

leads to the 1-norm on $\mathcal{P}^N$. Therefore, as long as $t_1(\mathbf{x})$ can be computed from $\mathbf{z}$, then the exponential reference hypothesis

$$p(\mathbf{x}|H_0) = \prod_{i=1}^{N} e^{-x_i} = e^{-t_1(\mathbf{x})} \tag{14}$$

meets the conditions of Theorem 4.

Example 2: Let $\mathbf{x}$ have support everywhere in $\mathcal{R}^N$. The statistic

$$t_2(\mathbf{x}) = \sum_{i=1}^{N} x_i^2 \tag{15}$$

leads to the 2-norm on $\mathcal{R}^N$. Therefore, as long as $t_2(\mathbf{x})$ can be computed from $\mathbf{z}$, then the manifold (6) will be inscribed on the hyper-sphere of radius $\sqrt{t_2(\mathbf{x})}$. The Gaussian reference hypothesis

$$p(\mathbf{x}|H_0) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}} e^{-x_i^2/2} = (2\pi)^{-N/2}\ e^{-t_2(\mathbf{x})/2}, \tag{16}$$

meets the conditions of Theorem 4.

D. Uniform Manifold Sampling (UMS) and Manifold Centroid.

The central idea of this paper follows by applying corrolary 1 to Theorems 3 or 4, which results in the following corrolary.

Corrolary 2: (Sampling the MaxEnt projected PDF). To draw samples from $G^*(\mathbf{x};T,g)$, the MaxEnt member of $\{\mathcal{P}:T,g\}$, we first draw a sample $\mathbf{z}^*$ from $g(\mathbf{z})$, then draw a sample $\mathbf{x}$ uniformly distributed on the manifold (6). To reflect this, the manifold distribution (7) simplifies to

$$\mu(\mathbf{x}|\mathbf{z}^*;T) = \frac{1}{\int_{\mathbf{x}\in\mathcal{M}(\mathbf{z}^*;T)}\mathrm{d}\mathbf{x}}, \quad \mathbf{x}\in\mathcal{M}(\mathbf{z}^*;T),\ 0\ \text{otherwise}, \tag{17}$$

where we have dropped the dependence on H_0.

Notice that we have removed the dependence of μ on H_0 because the reference distribution plays no role in the MaxEnt sampling procedure - the manifold distribution is always uniform and does not depend on H_0.

The second step, sampling from $\mu(\mathbf{x}|\mathbf{z};T)$, we call uniform manifold sampling (UMS), is useful by itself, and is related to feature inversion. Because the feature measures only a low-dimensional aspect of the original $\mathbf{x}$, the "inverted" $\mathbf{x}$ samples may appear entirely unlike the original $\mathbf{x}$ that led to $\mathbf{z}^*$. Therefore, in inverse problems, it is common to seek a "smooth" or optimum member of the manifold with respect to some "regularity" measure [12], [13]. UMS suggests a new approach to the inversion problem for convex manifolds: find the manifold centroid

$$\bar{\mathbf{x}}_z = \mathrm{E}(\mathbf{x}|\mathbf{z}^*) = \left(\int_{\mathbf{x}\in\mathcal{M}(\mathbf{z}^*;T)} \mathbf{x}\ \mathrm{d}\mathbf{x}\right) \Big/ \left(\int_{\mathbf{x}\in\mathcal{M}(\mathbf{z}^*;T)} \mathrm{d}\mathbf{x}\right), \tag{18}$$

which can be approximated by averaging independent samples generated by UMS for a fixed $\mathbf{z}$. As we will show, the centroid has very desirable properties in two important examples.

E. Chain Rule

Many feature transformations can be decomposed into a series of simpler transformations. What applies to the full transformation can be applied to each individual stage. But, interestingly, some transformations are only tractable when viewed as chains, and so it is only practical to do MaxEnt PDF projection on the chain.

Assume that the transformation $\mathbf{z} = T(\mathbf{x})$ can be broken into the parts $\mathbf{y} = T_y(\mathbf{x})$, $\mathbf{w} = T_w(\mathbf{y})$, and $\mathbf{z} = T_z(\mathbf{w})$. Then, equation (4) takes on the chain-rule form:

$$G(\mathbf{x}) = \left[\frac{p(\mathbf{x}|H_{0x})}{p(\mathbf{y}|H_{0x})}\right]\left[\frac{p(\mathbf{y}|H_{0y})}{p(\mathbf{w}|H_{0y})}\right]\left[\frac{p(\mathbf{w}|H_{0w})}{p(\mathbf{z}|H_{0w})}\right]\ g(\mathbf{z}), \tag{19}$$

where H_{0x}, H_{0y}, H_{0w} are reference hypotheses used at each stage. To understand the importance of the chain-rule, consider how we would compute $p(\mathbf{z}|H_0;T)$ for some canonical input data reference hypothesis $p(\mathbf{x}|H_0)$. At each stage, the distribution of the output feature becomes more and more intractable. Thus, at the end of a long signal processing chain, we may be unable to derive $p(\mathbf{z}|H_0;T)$. Estimating $p(\mathbf{z}|H_0;T)$ is futile, because generally a canonical reference hypotheses, as required for MaxEnt PDF projection is completely unrealistic as PDF for real data, and vice-versa. Furthermore, $p(\mathbf{z}|H_0;T)$ is more often than not evaluated in the far tails of the distribution. On the other hand, using the chain-rule, we can "re-start" the process by assuming a suitable canonical form for H_0 at the start of each stage. As long as each stage-dependent reference hypothesis meets the requirements for MaxEnt PDF projection for each stage by itself, then the chain as a whole will indeed produce the desired MaxEnt projected PDF [4]. This is most easily accomplished by including the appropriate energy statistic in the feature so that (11) and (12) can be satisfied, using canonical Gaussian, exponential, or uniform reference distributions.

To perform MaxEnt sampling from the chain (19), we first draw a sample $\mathbf{z}^*$ from $g(\mathbf{z})$. We then draw a sample $\mathbf{w}^*$ by UMS from the manifold $\{\mathbf{w} : T_z(\mathbf{w}) = \mathbf{z}^*\}$, then draw a sample $\mathbf{y}^*$ by UMS from the manifold $\{\mathbf{y} : T_w(\mathbf{y}) = \mathbf{w}^*\}$, then draw a sample $\mathbf{x}^*$ by UMS from the manifold $\{\mathbf{x} : T_y(\mathbf{x}) = \mathbf{y}^*\}$.

F. Remainder of the paper

Above, we have provided our main theoretical results. The rest of the paper is devoted to examining commonly-used feature transformations and describing how to draw samples from $G^*(\mathbf{x}; T, g)$. It is important to note that the simple examples provided below can be combined into chains (as per Section II-E) to analyze sophisticated feature transformations. We also will spend considerable time discussing the manifold centroid, which we alluded to in Section II-D. Not only does the centroid produce an interesting feature inversion approach, which has potential wide applications, but it is instrumental in speeding up the sampling itself.

III. UMS for Simple Feature Transformations

A. Magnitude Squared DFT bins

A non-linear feature transformation extensively used in signal processing is the DFT, followed by magnitude-squared of the bins. Let $\mathbf{x} \in \mathcal{R}^N$, where N is even. Let

$$z_k = \left| \sum_{i=1}^{N} x_i \; e^{-j2\pi(k-1)(i-1)/N} \right|^2, \quad 1 \le k \le N/2+1.$$

The computing of the projected PDF for this transformation was discussed previously (see [4], Section IV.A). The ES is the weighted total power $t_2(\mathbf{x}) = \sum_{k=1}^{N/2+1} z_k w_k$, where $w_k = 1/N$ for $k = 1, N/2+1$, and $w_k = 2/N$ otherwise. By Parseval's theorem, this computes the total input energy, $t_2(\mathbf{x}) = \sum_{i=1}^{N} x_i^2$, and therefore leads to the 2-norm $\|\mathbf{x}\|_2 = \sqrt{t_2(\mathbf{x})}$, satisfying (11). We can meet (12) with the Gaussian (16). Given a fixed feature $\mathbf{z}^*$, we generate samples of $\mathbf{x}$ on the manifold (6). We can treat each DFT bin independently. Note that $\mathbf{x}$ and the DFT output X_k, $1 \le k \le N/2+1$ are related by a linear transformation which preserves uniform distributions [2]. Thus, we may first generate the DFT output with UMS, then inverse transform to get $\mathbf{x}$. UMS for the real-valued DFT bins 1 and $N/2+1$ is accomplished by selecting X_k from $\{\sqrt{z_k^*}, -\sqrt{z_k^*}\}$ with equal probability. For the remaining bins, $X_k = \sqrt{z_k^*}\; e^{j\theta}$, where θ is selected uniformly in $[0,\ 2\pi]$. To compute $\mathbf{x}$, we extend the DFT output to length N using the conjugate of the complex bins, then take the inverse DFT.

B. Linear Feature, unbounded $\mathbf{x}$

1) Feature Transformation: Consider the linear feature calculation $\mathbf{z}_A = \mathbf{A}'\mathbf{x}$, where $\mathbf{x} \in \mathcal{R}^N$, and $\mathbf{A}$ is any full-rank $N \times D$ matrix. Applications include principal component analysis (PCA) and linear filtering of seismic recordings and time-series and the final DCT stage in computation of MFCC. This transformation has no ES, so needs to be augmented. Without loss of generality, we use (15). The complete feature is the union of $\mathbf{z}_A$ with the ES, denoted by $\mathbf{z} = [\mathbf{z}_A,\ t_2]$, which is a non-linear function of $\mathbf{x}$ and leads to a non-convex manifold.

2) UMS for Linear feature, unbounded $\mathbf{x}$: Given a fixed feature value $\mathbf{z}^* = [\mathbf{z}_A^*,\ t_2^*]$, the manifold is given by $\mathbf{x} : \left\{\mathbf{A}'\mathbf{x} = \mathbf{z}_A^*,\ \sum_{i=1}^{N} x_i^2 = t_2^*\right\}$. If $\mathbf{A}$ is full rank, then by orthogonal expansion, any $\mathbf{x}$ can be written

$$\mathbf{x} = \mathbf{x}_A + \mathbf{B}\mathbf{u}, \tag{20}$$

where $\mathbf{B}$ is the $N \times (N-D)$ ortho-normal matrix that spans the linear subspace orthogonal to the columns of $\mathbf{A}$, and

$$\mathbf{x}_A = \mathbf{A}(\mathbf{A}'\mathbf{A})^{-1}\mathbf{z}_A^*. \tag{21}$$

To satisfy the second requirement, we need $||\mathbf{x}||^2 = ||\mathbf{x}_A||^2 + ||\mathbf{u}||^2 = t_2^*$. Thus, we need the vector $\mathbf{u}$ to have length $||\mathbf{u}|| = \sqrt{t_2^* - ||\mathbf{x}_A||^2}$. Thus, $\mathbf{u}$ lies on a hyper-sphere. The multivariate standard Gaussian has a distribution that projects evenly anywhere on the standard hyper-sphere. Thus, uniformly sampling a hyper-sphere is accomplished by drawing $\mathbf{u}$ as $n-D$ *iid* samples of zero-mean Gaussian random variable, then normalizing $\mathbf{u}$ to have length equal to the hyper-sphere radius. In summary, the sampling method is: (a) Draw $N-D$ samples of independent Gaussian samples of mean 0 and variance 1, denoted by $\tilde{\mathbf{u}}$, (b) Let $\mathbf{u} = \frac{\tilde{\mathbf{u}}}{\|\tilde{\mathbf{u}}\|}\ \sqrt{t_2^* - ||\mathbf{x}_A||^2}$, then (c) Compute $\mathbf{x}$ from (20).

3) Asymptotic (large N) behavior: We generated random samples on the manifold for a fixed $\mathbf{z}^*$ and plotted the pair (x_l, x_m) for two indexes l, m on a plane. We used the feature $\mathbf{z} = [t_1(\mathbf{x}),\ t_2(\mathbf{x})]$, with $\mathbf{A} = [1,\ 1,\ \ldots 1]'$, and $D = 1$. Figure 2 shows a simulation for $N = 3, 4$, and 64. To compute $\mathbf{z}^*$, we drew a sample $\mathbf{x}$ from a standard normal distribution, computed t_1^*, t_2^*. Then, we generated random samples of $\mathbf{x}$ on the manifold. On the left, we see random samples of $\mathbf{x}$ projected on the x_1, x_2 plane. On the right, we see a histogram of x_2. Interestingly, for a 2-dimensional manifold, $N-D = 2$, the samples fall on an elliptical curve. With increasing manifold dimension, the marginal distributions approach Gaussian, even though the sampling is uniform on the manifold.

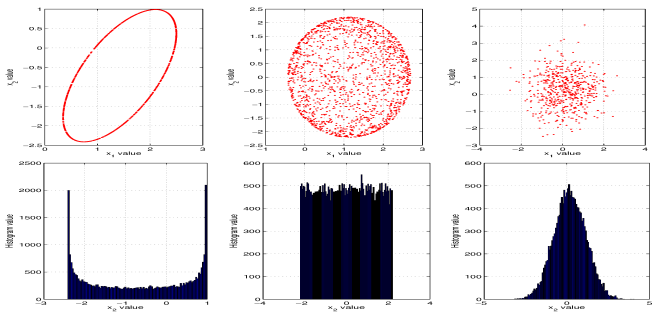

Fig. 2. Manifold sampling results for $N = 3$, $N = 4$, and $N = 64$. Top row: random samples of x_1, x_2. Bottom row: histograms of x_2.

[2]Note that the complex DFT operation, when the real and imaginary parts are concatenated, dimension-preserving, 1:1 and invertible real linear transformation. Under such transformations, the uniformly-distributed unit hypercube becomes rotated in higher-dimensional space, but remains uniformly distributed.

C. Maximum Likelihood (ML) Parameter Estimation

PDF projection can also be applied to iterative feature transformations ([4], Section II.C). We now consider UMS to generate samples uniformly on the manifold of data samples which will have exactly a specified ML parameter estimate. Consider the Gaussian parametric model $\mathbf{x} = \mathbf{H}_\theta \mathbf{a} + \mathbf{v}$, where $\mathbf{H}_\theta$ is a $N \times P$ matrix of basis functions, $\mathbf{a}$ is a $P \times 1$ amplitude vector, and $\mathbf{v}$ is an $N \times 1$ vector of *iid* zero-mean Gaussian RVs with variance σ^2. Assume column i of $\mathbf{H}_\theta$ depends non-linearly on a parameter θ_i. Let $\boldsymbol{\theta} = [\theta_1 \dots \theta_P]$. This model underlies many important estimation problems, such as the estimation of sine-waves in noise [14].

The likelihood function is given by

$$\log p(\mathbf{x}; \boldsymbol{\theta}, \mathbf{a}, \sigma^2) = -\frac{N}{2} \log(2\pi\sigma^2) - \frac{(\mathbf{x} - \mathbf{H}_\theta \mathbf{a})'(\mathbf{x} - \mathbf{H}_\theta \mathbf{a})}{2\sigma^2}.$$

The ML parameter estimates (MLE) $\hat{\boldsymbol{\theta}}, \hat{\mathbf{a}}, \hat{\sigma}^2$ are such that the derivative of $\log p(\mathbf{x}; \boldsymbol{\theta}, \mathbf{a}, \sigma^2)$ with respect to each parameter is zero. The derivative constraint for $\boldsymbol{\theta}$ leads to

$$\mathbf{x}' \mathbf{H}_{\hat{\theta}}^{\theta} \mathbf{a} = \mathbf{a}' \mathbf{H}_\theta' \mathbf{H}_{\hat{\theta}}^{\theta} \mathbf{a}. \tag{22}$$

where $\mathbf{H}_\theta^\theta$ is the column-wise derivative of $\mathbf{H}_\theta$ with respect to θ_i. Note that in these problems, $\hat{\boldsymbol{\theta}}$ can be found independently of $\hat{\mathbf{a}}$, $\hat{\sigma^2}$ by maximizing $\mathbf{x}' \mathbf{H}_\theta \left(\mathbf{H}_\theta' \mathbf{H}_\theta\right)^{-1} \mathbf{H}_\theta' \mathbf{x}$. Once $\hat{\boldsymbol{\theta}}$ is found, we have

$$\hat{\mathbf{a}} = \left(\mathbf{H}_{\hat{\theta}}' \mathbf{H}_{\hat{\theta}}\right)^{-1} \mathbf{H}_{\hat{\theta}}' \mathbf{x}, \tag{23}$$

then,

$$\hat{\sigma}^2 = \frac{1}{N} (\mathbf{x} - \mathbf{H}_{\hat{\theta}} \hat{\mathbf{a}})' (\mathbf{x} - \mathbf{H}_{\hat{\theta}} \hat{\mathbf{a}}). \tag{24}$$

Now, given fixed MLE values of $\hat{\boldsymbol{\theta}}$, $\hat{\mathbf{a}}$, $\hat{\sigma^2}$, the above equations can be used to define the manifold of $\mathbf{x}$ that lead to the given MLE. Define the statistics $\mathbf{z}_0 = \mathbf{x}'\mathbf{x}$, $\mathbf{z}_1 = \mathbf{H}_\theta' \mathbf{x}$, and $\mathbf{z}_2 = \hat{\mathbf{a}}' \mathbf{H}_\theta^{\theta\prime} \mathbf{x}$. We can use (23), (24), and (22) to compute $\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2$ from $\hat{\boldsymbol{\theta}}$, $\hat{\mathbf{a}}$, $\hat{\sigma^2}$. The equations defining $\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2$ lead to a set of constraints on $\mathbf{x}$ that can be written $\mathbf{A}'\mathbf{x} = [\mathbf{z}_1, \mathbf{z}_2]$, and $\mathbf{x}'\mathbf{x} = \mathbf{z}_0$, conforming to the problem of Section III-B. If we draw samples on the manifold, every sample will meet the derivative constraint for $\boldsymbol{\theta}$ as well as produce the same amplitude and variance estimates, so will be samples that produce the given ML solution.

IV. Linear feature, positive $\mathbf{x}$

We now study UMS for linear dimension-reducing transforms of positive data, with no upper bound in amplitude, thus $\mathbf{x} \in \mathcal{P}^N$. Applications include linear transformations of intensity or spectra.

A. Feature transformation

We consider again the linear feature calculation

$$\mathbf{z} = \mathbf{A}'\mathbf{x}, \tag{25}$$

where $\mathbf{A}$ is any full-rank $N \times D$ matrix. The ES (13) can be integrated into $\mathbf{z}$ if we assume that $\mathbf{1} = \mathbf{A} \left(\mathbf{A}'\mathbf{A}\right)^{-1} \mathbf{A}'\mathbf{1}$, where $\mathbf{1} = [1, 1, 1, \dots 1]'$.

B. UMS for Linear Feature, positive $\mathbf{x}$

For a fixed feature $\mathbf{z}^*$, UMS is implemented by drawing a sample from the manifold $\{\mathbf{x} : \mathbf{A}'\mathbf{x} = \mathbf{z}^*, \quad \mathbf{x} \in \mathcal{P}^N\}$. We can span the manifold by choosing $\mathbf{u}$ in (20). Samples may be generated using rejection sampling by generating samples of $\mathbf{u}$ uniformly in a sufficiently large hypercube, then rejecting samples $\mathbf{x}$ that fall outside of $\mathcal{P}^N$, but this method suffers from exponentially decreasing acceptance rate [15].

To visualize the distribution of $\mathbf{x}$ generated using UMS, we conducted an experiment analogous to section III-B3. We used a feature of dimension $D = 2$, with the first feature equal to $t_1(\mathbf{x})$, and generated random samples of $\mathbf{x}$ on the manifold using rejection sampling. Figure 3 (top left) shows samples of x_1, x_2 showing the desired uniform distribution. Figure 3 (top right) shows the histogram of x_2. For manifold dimensions

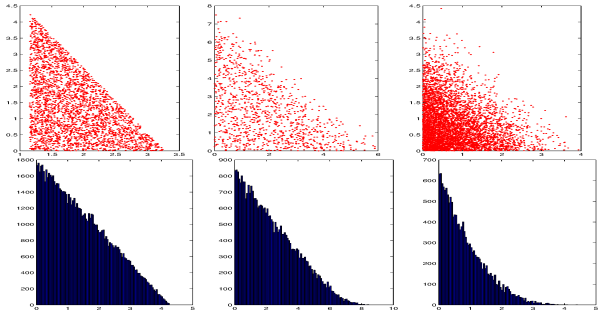

Fig. 3. Manifold sampling results for $N = 4$, $N = 6$, and $N = 10$ (manifold dimension 2,4,8, respectively). Top row: random samples of x_1, x_2. Bottom row: marginal distribution of x_2.

above 2, the manifold distribution does not look uniform when projected onto a 2D plane even though it is uniform in the higher dimensions. With increasing manifold dimension, the marginal distribution looks increasingly exponential. This effect is analogous to Figure 2, which tended to Gaussian. It is interesting that the marginal distributions, despite using a uniform distribution on the manifold, tend to canonical distributions, which are known to be maximum entropy distributions under the moment constraints corresponding to the respective energy statistics [16], [11].

C. MCMC-UMS

1) Hit-and-Run: To make UMS practical at high dimensions, we turn to a form of MCMC called Hit-and-Run (H&R) for uniformly sampling on a compact convex set [17], [18], [19], [15].

2) Algorithm description: In MCMC-UMS, we first find one valid sample on the manifold, then move along a 1-dimensional line in the linear subspace spanned by the columns of matrix $\mathbf{B}$ which is defined in (20). Since the manifold set is convex, all points on the line in a compact interval are valid and the interval endpoints can be easily solved for. A new sample is chosen by uniformly sampling on the interval. The process repeats by choosing a new direction, and repeating. This is considered to be the most efficient way to obtain an asymptotic uniformly-sampled point in a convex set [17]. We consider two ways of choosing directions:

1) *Random-directions* (normal H&R) in which a new random direction is chosen each time. We choose the new direction by drawing a vector of independent Gaussians, and normalizing it to have norm 1.
2) *Systematic* (variation of H&R), by changing each element of $\mathbf{u}$ one at a time, a method can be seen as slice sampling in multiple dimensions with uniform target distribution [20].

Here is a summary of the MCMC-UMS algorithm.

1) As a starting point, we will need a vector $\mathbf{u}$ that is valid, i.e. produces an $\mathbf{x}$ that lies in $\mathcal{M}(\mathbf{z}^*;T)$ and $\mathcal{P}^N$. Let $\mathbf{x}^0$ be a starting point, then $\mathbf{u}^0 = \mathbf{B}'\mathbf{x}^0$, which maps back to $\mathbf{x}^0$ using (20).
2) For *systematic method*, we let $\mathbf{b}$ be a column of matrix $\mathbf{B}$. Each iteration, we choose the next column, in order, returning back to the first column after $N-D$ iterations. In the *random directions* method, we choose $\mathbf{b}$ to be a random direction within the column space of $\mathbf{B}$ by generating an $N-D$-dimensional vector $\mathbf{h}$ of independent standard normal variates, normalized so that $\|\mathbf{h}\| = 1$. We then let $\mathbf{b} = \mathbf{B}\mathbf{h}$.
3) We then let $\mathbf{x} = \mathbf{x}^0 + \gamma\mathbf{b}$, where γ lies in a compact interval of the real line $\gamma \in (\gamma^L,\ \gamma^H)$, that includes zero (to produce the current value $\mathbf{x}^0$). We find these limits as follows. Define the vector $\mathbf{c} = [c_1, c_2, \ldots c_N]$ calculated from $\mathbf{b}$ as $c_i = b_i/x_i^0, \forall i$. Then, γ^L is -1 times the reciprocal of the largest positive value of $\mathbf{c}$, and γ^H is -1 times the reciprocal of the most negative value of $\mathbf{c}$.
4) We select γ by drawing a uniform random variable in $(\gamma^L,\ \gamma^H)$.
5) We then set $\mathbf{x}^0 = \mathbf{x}$, and repeat (go to step 2).

The number of iterations required before the initial conditions are "forgotten" is problem-dependent.

3) Starting point: To start MCMC-UMS, we need a valid $\mathbf{x}$ that is a solution to $\mathbf{A}'\mathbf{x} = \mathbf{z}^*,\ \mathbf{x} \in \mathcal{P}^N$. A good starting point can be obtained from any linear-programming solver by finding the solution to the maximization [3] of $q = \sum_{i=1}^N x_i$ subject to $\mathbf{A}'\mathbf{x} = \mathbf{z}^*$. Any linear programming (LP) algorithm such as OCTAVE `glpk.m` or MATLAB `linprog.m` can output a "solution" which is a valid point. However, better than a starting point from a linear programming solver is the manifold centroid described later in Section IV-D.

4) Illustration: To illustrate the method, we choose $N = 5$, and $D = 3$ so that $N - D = 2$ so that the data will always appear uniformly distributed when projected on a plane. Figure 4 shows the data as seen from the x_2, x_5 plane after 4 and 20 iterations of the systematic method. The five facets of the valid region are caused by the manifold reaching the positivity limit of each of the five dimensions. Non-uniformity can be clearly seen for 4, but not for 20 iterations. Later, we will discuss the difference between the systematic and the random-directions algorithm. But first, we must solve for the centroid.

[3]It makes no difference if we are maximizing or minimizing since q is fixed anyway by the linear constraints.

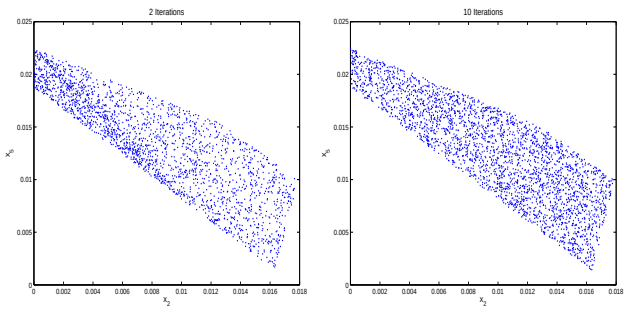

Fig. 4. Visual uniformity test for $N = 5$, $D = 3$. The 2-D manifold as seen from the two dimensions x_2, x_5 after 4 iteration (left) and 20 iterations (right).

D. Centroid

For convex manifolds, the manifold centroid $\bar{\mathbf{x}}_z$ is the center of mass in Figure 4 and is on the manifold. It is also the conditional mean $\mathcal{E}(\mathbf{x}|\mathbf{z}^*)$, and is an optimal point with respect to a deterministic entropy measure. It has applications in feature inversion, image reconstruction and spectral estimation.

The centroid can be approximated by the sample mean of samples generated using UMS, or much more efficiently using the "surrogate density" approach, which we now explain. Let $p_s(\mathbf{x})$, be a PDF with support on $\mathcal{X}$ (not limited to the manifold), but sharing four properties with $\mu(\mathbf{x}|\mathbf{z}^*;T)$: (a) its mean λ lies on the manifold, so

$$\mathbf{A}'\lambda = \mathbf{z}^*, \quad \text{such that } \bar{x}_i > 0, \text{ for all } i, \tag{26}$$

(b) it has constant density *along* the manifold (meaning that the gradient in a direction aligned with the manifold is zero), (c) it has maximum possible entropy under the constraint (26). However, instead of having all its probability mass *on* the manifold, it has support in all of $\mathcal{X}$ with its probability mass concentrated *near* the manifold. This idea is illustrated in Figure 5. The property that the samples congregate near the manifold for N large can be justified by the law of large numbers (See Section VIII-A). As a result, the surrogate density converges effectively to the manifold distribution. Therefore, the mean $\boldsymbol{\lambda}$ of the surrogate density is a very good approximation to the manifold centroid $\bar{\mathbf{x}}_z$ at high dimensions. The property that the surrogate distribution is uniform along the manifold can be seen once we select the surrogate density and maximize its entropy. It is known that the exponential density has the highest entropy among all densities for positive-valued $\mathbf{x}$ with specified mean $\boldsymbol{\lambda}$ [11].

$$p(\mathbf{x};\boldsymbol{\lambda}) = \prod_{i=1}^N \frac{1}{\lambda_i}\ \exp\left\{-\frac{x_i}{\lambda_i}\right\}. \tag{27}$$

We therefore propose to use (27) as the surrogate density for $\mu(\mathbf{x}|\mathbf{z}^*;T)$, by maximizing the entropy of (27) over $\boldsymbol{\lambda}$, subject to $\mathbf{A}'\boldsymbol{\lambda} = \mathbf{z}^*$. The entropy of (27) is

$$Q_p = \sum_{i=1}^N (1 + \log\lambda_i), \tag{28}$$

where "p" indicates positive data case. If we use (20) to write $\boldsymbol{\lambda}$ in terms of $\mathbf{u}$, we can maximize Q_p over $\mathbf{u}$. The solution

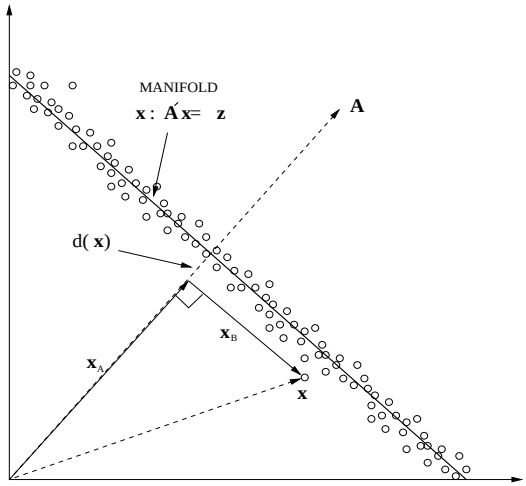

Fig. 5. Illustration of surrogate density. An arbitrary sample $\mathbf{x}$ is decomposed into a component $\mathbf{x}_A$ in the column space of $\mathbf{A}$ and the orthogonal component $\mathbf{x}_B$. At high dimension, samples congregate near the manifold where $\mathbf{A}'\mathbf{x} = \mathbf{z}$ and are equally distributed along the manifold.

must meet the requirement that the derivatives of the entropy with respect to u_k are zero, or

$$Q_p^{u_k} = \sum_{i=1}^{N} \frac{B_{ik}}{\lambda_i} = 0, \quad 1 \le k \le m. \tag{29}$$

This condition forces the distribution to be constant on the manifold. To see this, first, let $\mathbf{x}$ be decomposed as (See Figure 5), $\mathbf{x} = \mathbf{x}_A + \mathbf{B}\mathbf{u}$, where matrix $\mathbf{B}$ spans the subspace orthogonal to matrix $\mathbf{A}$. Note that changes to vector $\mathbf{u}$ will move $\mathbf{x}$ within the manifold, but not change its projection onto the columns of $\mathbf{A}$, so $\mathbf{x}$ remains on the manifold. Therefore, a distribution is constant on the manifold if and only if its derivative w/r to $\mathbf{u}$ is zero. It is easily shown that the derivative of $\log p(\mathbf{x};\boldsymbol{\lambda})$ with respect to u_k equals $-\sum_{i=1}^{N} \frac{B_{i,k}}{\lambda_i}$, making (29) equivalent to requiring $p(\mathbf{x};\boldsymbol{\lambda})$ to be constant on the manifold. Now we see that the surrogate density not only has mean value on the manifold, has probability mass congregating near the manifold, and in addition is constant on the manifold. Therefore, we must also expect that it has asymptotically the same mean as the manifold distribution (17).

Incidentally, note that maximizing (28) also maximizes spectral entropy measure

$$Q_s(\mathbf{x}) = \sum_{i=1}^{N} \log x_i, \tag{30}$$

which underlies classical MaxEnt spectral estimation [21], [22] and MaxEnt image reconstruction [23], [24]. Thus, we expect to obtain the same result as classical methods.

Note that (29) is the same as the condition that the vector

$$\boldsymbol{\alpha} = [1/\lambda_1, 1/\lambda_2 \ldots 1/\lambda_N]' \tag{31}$$

is contained in the column space of $\mathbf{A}$. Therefore, we can replace (29) with the alternative condition

$$\boldsymbol{\alpha} = \mathbf{A}\mathbf{v}, \tag{32}$$

for some $D \times 1$ vector $\mathbf{v}$. Therefore, to find the mean of the surrogate density, we solve for the free variable $\mathbf{v}$ such that

$$\mathbf{A}'\boldsymbol{\lambda}(\mathbf{A}\mathbf{v}) = \mathbf{z}^*, \tag{33}$$

where

$$\boldsymbol{\lambda}(\boldsymbol{\alpha}) = [1/\alpha_1, 1/\alpha_2, \ldots 1/\alpha_N]'. \tag{34}$$

We can solve this by driving the square error to zero:

$$\rho(\mathbf{v}) = (\mathbf{A}'\boldsymbol{\lambda}(\mathbf{A}\mathbf{v}) - \mathbf{z})'(\mathbf{A}'\boldsymbol{\lambda}(\mathbf{A}\mathbf{v}) - \mathbf{z}). \tag{35}$$

We can easily find derivative of $\rho(\mathbf{v})$ with respect to $\mathbf{v}$:

$$\left[\frac{\partial \rho}{\partial v_k}\right] = 2(\mathbf{A}'\boldsymbol{\lambda}(\mathbf{A}\mathbf{v}) - \mathbf{z})'\mathbf{A}'\boldsymbol{\Lambda}\mathbf{A}, \tag{36}$$

where $\boldsymbol{\Lambda}(\boldsymbol{\alpha})$ is the diagonal matrix with diagonal elements

$$\Lambda_i = -1/\alpha_i^2. \tag{37}$$

We have found that if we use the negative-definite Hessian approximation

$$\left[\frac{\partial^2 \rho}{\partial v_k \partial v_l}\right] \simeq -2\left(\mathbf{A}'\boldsymbol{\Lambda}(\boldsymbol{\alpha})\mathbf{A}\right)\left(\mathbf{A}'\boldsymbol{\Lambda}(\boldsymbol{\alpha})\mathbf{A}\right), \tag{38}$$

the resulting Newton-Raphson algorithm has excellent convergence properties when starting with $\boldsymbol{\alpha} = [1, 1, \ldots 1]'$. Let $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$ be the value of $\boldsymbol{\lambda}$ at the solution to (33). The algorithm to find $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$ can be summarized as follows.

1) Set iteration counter $n = 0$.
2) To initialize, let $\mathbf{v}_0 = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{1}$, where $\mathbf{1}$ is the vector of ones.
3) $\boldsymbol{\alpha}_n = \mathbf{A}\mathbf{v}_n$. Initially, $\boldsymbol{\alpha}$ will be the vector of ones.
4) $\boldsymbol{\lambda}_n(\boldsymbol{\alpha}_n) = [1/\alpha_1, 1/\alpha_2, \ldots 1/\alpha_N]'$.
5) Compute derivative and Hessian according to (36),(37) and (38), then update $\mathbf{v}$:

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \left[\frac{\partial^2 \rho}{\partial u_k \partial u_l}\right]^{-1}\left[\frac{\partial \rho}{\partial u_k}\right].$$

6) Increment n and go to step 3.

Although this method corresponds to classical methods, it is based on a novel sampling argument and can be extend to other manifolds as we will see.

E. Example: AR/ACF Spectrum

We now use UMS to create random spectra that have a fixed auto-correlation function (ACF) up to a lag of 2, corresponding to an auto-regressive (AR) process of order $P = 2$ [25].

1) Feature Transformation: Let $\mathbf{x}$ be the $N \times 1$ vector of magnitude-squared bins of the DFT of N_t real time-series samples, where $N = N_t/2 + 1$. The ACF can be computed by the inverse DFT using $\mathbf{z} = \mathbf{A}'\mathbf{x}$, where matrix $\mathbf{A}$ is described in ([4], Section V.B). The ES is the zero-th ACF lag, which is a weighted sum of the elements of $\mathbf{x}$, with corresponding reference hypothesis ([4], equation 28).

2) Experimental approach: To generate a realistic input sample $\mathbf{x}$, we generated simulated time-series data of length $N_t = 128$ from an AR process of order $P = 2$ with a peaked spectrum. We then computed the magnitude-squared bins of the DFT, keeping the $N = 65$ unique bins as our "raw data" $\mathbf{x}$. Multiplying by the 65×3 matrix $\mathbf{A}$ computed the first three auto-correlation lags (0 through 2). With this feature fixed, we applied MCMC-UMS (systematic approach) to generate random spectra on the manifold. We compared the mean of the MCMC-UMS generated samples with $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$.

3) Results: Figure 6 shows five spectra: (a) input spectrum $\mathbf{x}$, (b) a typical MCMC-UMS sample, (c) traditional auto-regressive (AR) spectrum obtained by Levinson algorithm, (d) MaxEnt solution $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$ solving (29), and (e) the sample mean of 100000 full MCMC-UMS iterations[4]. All of the spectra are on the manifold $\mathcal{M}(\mathbf{z}^*; T)$ [5]. The important take-away from Figure 6 is that $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$ runs directly through the MCMC-UMS sample mean values (circles), confirming that $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$ is indeed a good estimate of $\bar{\mathbf{x}}_z$. The fact that $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$ also appears to be the same as the traditional AR spectrum (aside for a slight deviation at the end bins), is not unexpected because it involves the same optimization as for traditional MaxEnt spectral estimation, which is known to coincide with the auto-regressive method [21], [22]. Despite the fact that $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$ corresponds to classical methods in this case, it is enlightening to see a new geometric interpretation at the manifold centroid and maximum entropy sampling interpretation. It also leaves open the possibility of generalizing to different linear constraints and different range for $\mathbf{x}$.

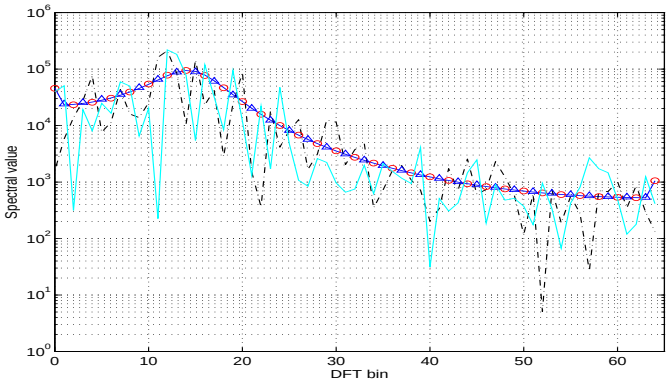

Fig. 6. Light jagged line: raw input spectrum $\mathbf{x}$. Dark jagged dashed line: Typical MCMC-UMS sample. Circles: sample mean of MCMC-UMS. Triangles: traditional AR spectrum. Smooth dark solid line that runs through circles and triangles: MaxEnt solution $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$.

F. Accuracy of Estimated Centroid and Mixing Rate

While Figure 6 suggests that $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*) \simeq \bar{\mathbf{x}}_z$, it is prudent to quantify the accuracy. We can visualize the convergence of the sample mean to the centroid estimate, by plotting the error metric

$$\epsilon = \frac{1}{N} \sum_{i=1}^{N} \left[\log \left(\frac{\bar{x}_i}{\hat{\lambda}_i} \right) \right]^2 , \tag{39}$$

where $\bar{x}_i$ is the sample mean of x_i and λ_i is corresponding element of $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$. In Figure 7, we plot mean square fractional error ϵ as a function of *full iterations*[6] for the systematic and random directions approaches. The graph also shows the result with whitening, which will be described later.

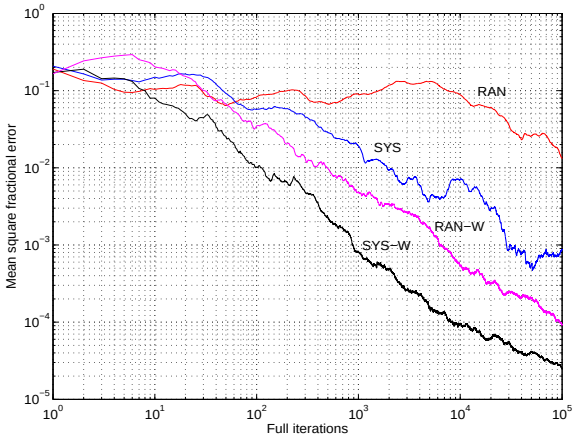

Fig. 7. Convergence of MCMC-UMS sample mean to MaxEnt solution for four approaches: systematic (SYS), random directions (RAN), with whitening (-W). X-axis: number of full iterations.

The more correlated the samples in consecutive iterations are, the longer it takes for the sample mean to converge. The mixing rate for MCMC is generally measured by effective sample size (ESS), which is the number of independent samples needed to obtain the equivalent estimation error [26]. We can see the change in ESS in Figure 7 by drawing horizontal lines. The error continues decreasing even after 100,000 full iterations, an indication that the proposed MaxEnt solution is a good approximation to the asymptotic mean of MCMC-UMS.

Figure 7 surprisingly indicates that the random directions method is much slower. To investigate this, we tried the systematic approach after randomizing the orthogonal basis matrix $\mathbf{B}$, by post-multiplying it by an orthogonalized random matrix of dimension $(N - D) \times (N - D)$. Surprisingly, the mixing rate then matched that of random directions. It appears, then, that the advantage of the systematic method comes not from the systematic way that the basis functions are used, but has to do with the fact that the basis functions, as they are calculated by SVD or QR decomposition of matrix $\mathbf{A}$, are optimally aligned. Computing $\mathbf{B}$ from $\mathbf{A}$ using either SVD and QR had the same effect. So, the same orthogonal subspace is spanned by the randomized matrix $\mathbf{B}$ but has much slower mixing. To understand this effect, suppose we start at one end of a long, thin convex region. Moving in random directions, it is likely to take a long time to cover the whole region. But, if one of the systematic directions are aligned with the main axis,

[4]Each full iteration is $N - D = 62$ simple iterations, updating each of the free manifold dimensions.

[5]Except the AR spectrum because it only *approximately* meets the constraint. The ACF of the AR spectral estimate matches the ACF of the original spectrum [25], but this is a large-N asymptotic property that does not hold when the ACF is computed using a finite-length data segment

[6]A full iteration is a set of $N - D$ regular iterations, which completes the update of all free dimensions in the systematic approach

we can quickly cover the region. Why this optimal alignment happens and whether it is problem-specific is not clear.

G. Whitened MCMC-UMS

Since we have verified that $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*) \simeq \bar{\mathbf{x}}_z$, we can now estimate the centroid (mean of samples drawn using MCMC-UMS) and can "whiten" the problem. Let $\tilde{\mathbf{x}}$ be the "whitened" raw input spectrum, $\tilde{x}_i = x_i/\hat{\lambda}_i, \quad 1 \leq i \leq N$, and $\tilde{\mathbf{A}}$ the compensated matrix $\tilde{\mathbf{A}}_{i,j} = \mathbf{A}_{i,j}\hat{\lambda}_i, \ \forall i, \ \forall j$, which obtains the same feature $\mathbf{z} = \mathbf{A}'\mathbf{x} = \tilde{\mathbf{A}}'\tilde{\mathbf{x}}$. The new linear constraint "manifold" is:

$$\tilde{\mathbf{x}} : \ \tilde{\mathbf{A}}'\tilde{\mathbf{x}} = \mathbf{z}^*. \tag{40}$$

The vector of ones, denoted by $\mathbf{1}$, can be used as a valid starting vector since $\tilde{\mathbf{A}}'\mathbf{1} = \tilde{\mathbf{A}}'\hat{\boldsymbol{\lambda}}(\mathbf{z}^*) = \mathbf{z}^*$. MCMC-UMS will produce vectors with mean $\mathbf{1}$ and meet (40). These vectors can then be transformed using $x_i = \tilde{x}_i\hat{\lambda}_i, \quad \forall i$, to solve the original problem because they are in $\mathcal{M}(\mathbf{z}^*; T)$ and uniformly distributed [7]. The result of whitening can be seen in Figure 7 and the result is dramatic. The change in ESS (seen by drawing a horizontal line) and is more than an order of magnitude for both systematic and random directions.

H. Dimension effects

To see the effect of dimension on the accuracy of $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$ in predicting the centroid, we repeated the experiment of Figure 7 for various manifold dimensions $N - D$. In Figure 8, we see the value of ϵ after 20,000 full iterations, averaged over three trials (systematic approach with whitening). at each dimension. Even for manifold dimension 3, there is good accuracy: $\epsilon = .003$ means that the fractional error has a standard deviation of .055 or 5.5% error. At dimension 125, ϵ is about .00009, a fractional error standard deviation of 1%.

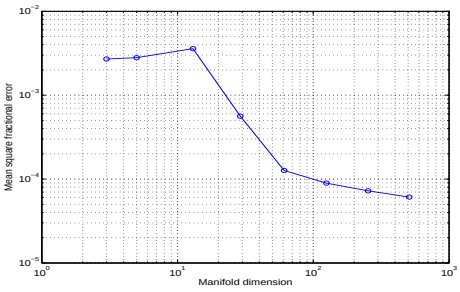

Fig. 8. Mean square fractional error as a function of dimension estimated using 20,000 iterations.

I. MFCC

In section IV-E3, we showed empirically that for ACF constraints, $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$ is related to the Burg maximum entropy spectral estimation. We now ask if the method can be extended to other linear constraints (other than ACF) and if this also results in good spectral estimators. The MEL frequency cepstral coefficients (MFCC) is the predominant feature extraction method used in human speech analysis [27], [28]. If we disregard the final stages of logarithm and discrete cosine transform (DCT), the MFCC features differ from ACF features only in the matrix $\mathbf{A}$. For MFCC, the columns of $\mathbf{A}$ are the MEL band functions, which are shown for $N_c = 24$ bands in Figure 9 for $N_t = 768$ and $N = 385$. Note that in Figure 9, the sum of the MEL band functions (line on top) is a constant, which shows that the requirement to contain an energy statistic is met.

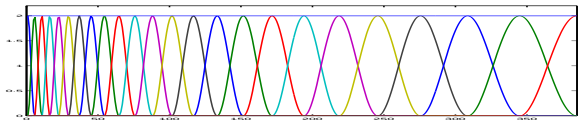

Fig. 9. MEL band functions for $N = 768$. There are 24 bands including the zero and Nyquist bands. Their sum, the flat line on top, is a constant.

In Figure 10, we see results of MCMC-UMS, similar to Figure 6. We calculated the MFCC features $\mathbf{z}^*$ from 768 samples of human speech at 12kHz sample rate. With the feature value fixed, we used MCMC-UMS to produce random spectra. We overlaid the initial LP solution, the MaxEnt solution $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$, and the average of 10000 full MCMC-UMS iterations ontop of one random MCMC-UMS sample. Again, we conclude that $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$ precisely estimates the centroid. Note that $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$ is a very smooth spectral estimate that is visually very satisfactory. Our proposed method may be preferred to open-ended MFCC synthesis methods [29] since the resulting spectrum is feature-reproducing and has optimal smoothness since it satisfies the maximum entropy rule, which is the same as maximizing the spectral flatness [23], [25].

J. Complete Chains

Based on the information presented in the previous paper [4] and what we presented here, we have provided a means to create complete generative models from the signal processing chains for the calculation of both AR and MFCC coefficients. Combined with a kernel-based PDF estimation of the features, either single or as a sequence using hidden Markov model (HMM), we can compute likelihood function values and produce unlimited samples.

For an AR-based generative model, the PDF projection was described in ([4], Section V.B). Sampling requires DFT/magnitude-squared (Section III-A), and ACF calculation (Section IV-E). For better-behaved features that lend themselves to modeling by kernel-based PDF estimators, we suggest additional 1:1 transformations converting to reflection coefficients ([30], Section VI.D.3), and finally log area ratio coefficients ([30], Section VI.D.4).

For MFCC, PDF projection was described in ([4], Section V.A). Sampling requires DFT/magnitude-squared (Section III-A), MEL band energy binning (Section IV-I), log transformation ([30], Section VI.A), and finally DCT with truncation of output coefficient set (an application of Section III-B).

V. Linear feature, doubly-bounded $\mathbf{x}$

We now treat UMS for linear transforms of data bounded in amplitude above and below (doubly-bounded) by adapting

[7] The linear whitening operation preserves the uniform distribution.

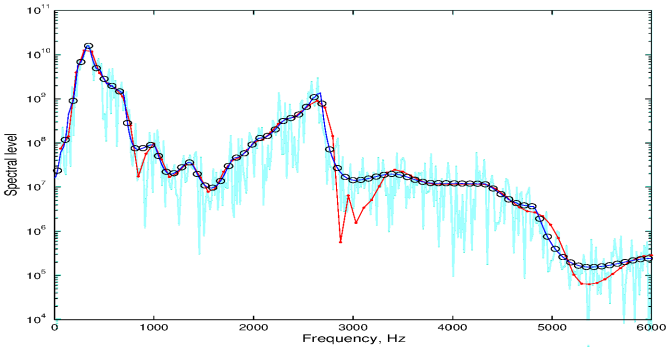

Fig. 10. Light jagged line: One sample spectrum from MCMC-UMS. Line with dots: LP solution, Circles: $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$, Curve through circles: average of 10000 MCMC-UMS samples.

the methods of Section IV, which were shown to maximize the classical spectral entropy (30). The solution to the doubly-bounded problem has no classical equivalent and results in a novel entropy measure. With no loss of generality, data is assumed to be in the range $\mathbf{x} \in \{\mathbf{x} : \ 0 < x_i < 1, \quad \forall i\}$ and the feature transformation is given by (25). Applications include PCA and linear transforms of doubly-bounded data including optical character recognition (OCR), and neural networks.

A. MCMC-UMS for doubly-bounded $\mathbf{x}$

Because the input space is doubly-bounded in all dimensions, it is compact. No energy statistic is needed (See Theorem 3). So, it is not necessary to insure that the 1-norm can be computed from $\mathbf{z}$, although including it may improve sampling and reconstruction of $\mathbf{x}$. The sampling procedure is very similar to the previous example, section IV. Generating data by MCMC-UMS is affected by the double bound on the input data and the H&R procedure detailed in Section IV-C2 is easily adapted. We calculate the bounds $\gamma^L, \ \gamma^H$ the same as before, which are related to the lower bound on the input data. We also calculate two additional bounds $\tilde{\gamma}^L, \ \tilde{\gamma}^H$, related to the upper bound on the input data. We define the vector $\mathbf{d} = [d_1, d_2, \ldots d_N]$ calculated from $\mathbf{b}$ as $d_i = b_i/(1 - x_i^0), \quad 1 \leq i \leq N$. Then, $\tilde{\gamma}^L$ is equal to the reciprocal of the most negative value of $\mathbf{d}$, and $\tilde{\gamma}^H$ is the reciprocal of the largest positive value of $\mathbf{d}$. Then, the lower bound on u_i is the largest of γ^L and $\tilde{\gamma}^L$, and the upper bound on u_i is the smallest of γ^H and $\tilde{\gamma}^H$.

B. Centroid

We now apply the method of Section IV-D to solve for the centroid in the doubly-bounded case by finding a suitable surrogate density. Recall that in Figures 2, and 3, when N becomes large, the marginal distributions of UMS samples approach the shape of the maximum entropy densities under the applicable constraints (Gaussian and exponential, respectively), which are the suitable surrogate densities. Figure 11 shows the marginal distribution of an element of the doubly-bounded input data for UMS samples which appears to be a truncated exponential distribution (TED) with positive exponent. In fact, for data bounded to the interval $[0, 1]$, the TED is the maximum entropy distribution under mean (first moment) constraints ([31], page 186). Let's take a closer

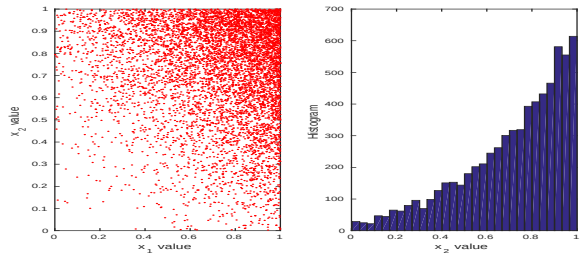

Fig. 11. From top: manifold sampling results using MCMC-UMS $N = 32$, plotting x_1, x_2. Right, histogram of x_2.

look at this distribution. For data x in the interval $[0, 1]$ and exponent parameter α (which can be positive or negative), the uni-variate truncated exponential distribution (TED) is $p(x; \alpha) = C(\alpha) \ e^{\alpha x}$, where $C(\alpha) = \left(\frac{\alpha}{e^{\alpha}-1}\right)$. The mean of this density is given by

$$\lambda(\alpha) = \int_0^1 x \ p(x; \alpha) \ \mathrm{d}x = \frac{e^{\alpha}}{e^{\alpha} - 1} - \frac{1}{\alpha} \tag{41}$$

and the entropy is given by

$$Q_{db}(\alpha) = -\log\left(\frac{\alpha}{e^{\alpha} - 1}\right) - \alpha\lambda(\alpha), \tag{42}$$

where "db" indicates the doubly-bounded case. The multivariate TED is

$$\log p(\mathbf{x}) = \sum_{i=1}^{N} \log C(\alpha_i) + \alpha_i x_i. \tag{43}$$

The entropy of the multivariate density of N independent truncated exponentials is therefore, for input data constrained to the unit interval $[0, 1]$ is

$$Q_{db}(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \left\{ -\log\left(\frac{\alpha_i}{e^{\alpha_i} - 1}\right) - \alpha_i \lambda_i \right\}, \tag{44}$$

where λ_i is computed from α_i using (41).

C. Solving for asymptotic mean of MCMC-UMS

Following the method of section IV-D, we propose to use (43) as the surrogate density. More precisely, we propose to maximize the entropy (44) over $\boldsymbol{\lambda}$ subject to $\mathbf{A}'\lambda = \mathbf{z}$. Unfortunately, the entropy is written in terms of both $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$. But, it can be shown that for a given λ, there is a unique α [32]. This is true in one or more dimensions. Therefore, $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$ are alternative parameterizations for the multivariate truncated exponential distribution.

In the same manner as in Sections III-B and IV, we use $\mathbf{u}$ as the free variable under the constraint (26). So, to maximize (44), we need the derivatives of (44) with respect to the elements of $\mathbf{u}$. Using the derivative chain-rule, we can write the first derivative of (44) with respect to u_i $Q_{db}^{u_i} = \sum_{k=1}^{N} Q_{db}^{\alpha_k} \left(\frac{\mathrm{d}\lambda_k}{\mathrm{d}\alpha_k}\right)^{-1} \frac{\partial \lambda_k}{\partial u_i}$, where, from (44) $Q_{db}^{\alpha_k} = -\frac{1}{\alpha_k} + \alpha_k \frac{e^{\alpha_k}}{(e^{\alpha_k}-1)^2}$. From (41) , $\frac{\mathrm{d}\lambda_k}{\mathrm{d}\alpha_k} = \frac{1}{\alpha_k^2} - \frac{e^{\alpha_k}}{(e^{\alpha_k}-1)^2}$.

And, from (20), $\frac{\mathrm{d}\lambda_k}{\mathrm{d}u_i} = B_{k,i}$. After (a lot of) cancellations, we get

$$Q_{db}^{u_i} = -\sum_{k=1}^{N} \alpha_k B_{k,i}, \tag{45}$$

resulting in the condition for maximization of entropy:

$$\sum_{k=1}^{N} \alpha_k B_{k,i} = 0, \quad \forall i. \tag{46}$$

But, note that (46) is equivalent to the statement that the vector $\boldsymbol{\alpha}$ is in the column space of $\mathbf{A}$, or that there exists a free variable $\mathbf{v}$ such that $\boldsymbol{\alpha} = \mathbf{A}\mathbf{v}$. As an aside, note that condition (46) also assures that (43) will have zero derivative along the manifold, which is one of the assumptions of the surrogate density.

The method to find the centroid is to find the vector $\mathbf{v}$ such that

$$\mathbf{A}'\lambda(\mathbf{A}\mathbf{v}) = \mathbf{z}^*, \tag{47}$$

where $\lambda(\boldsymbol{\alpha})$ is equation (41) applied element-wise. This is essentially the same as for the positive-$\mathbf{x}$ case (33) , except that the non-linear relationship between $\boldsymbol{\lambda}$ and $\boldsymbol{\alpha}$ is different. The algorithm of Section IV-D to find $\boldsymbol{\lambda}$ based on driving (35) to zero can be used if $\mathbf{v}_0 = \mathbf{0}$ and the diagonal elements of $\boldsymbol{\Lambda}$ in (36) are given by

$$\Lambda_i = \frac{1}{\alpha_i^2} - \frac{e^{\alpha_i}}{(e^{\alpha_i} - 1)^2}. \tag{48}$$

Let $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$ be the value of $\boldsymbol{\lambda}$ at the solution to (47). The modified algorithm to find $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$ is:

1) Set iteration counter $n = 0$.
2) To initialize, let $\mathbf{v}_0 = \mathbf{0}$.
3) $\boldsymbol{\alpha}_n = \mathbf{A}\mathbf{v}_n$. Initially, $\boldsymbol{\alpha}$ will be the vector of zeros.
4) Compute $\boldsymbol{\lambda}_n$ from $\boldsymbol{\alpha}_n$ using (41) element-wise.
5) Compute derivative and Hessian according to (36) and (38), and (48), then update $\mathbf{v}$:

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \left[\frac{\partial^2 \rho}{\partial u_k \partial u_l}\right]^{-1} \left[\frac{\partial \rho}{\partial u_k}\right].$$

6) Increment n and go to step 3.

D. Simple Example

In this experiment, we used a data size of $N = 128$. The matrix $\mathbf{A}$ in (25) computed the first $D = 6$ coefficients of the length-N DCT of $\mathbf{x}$. We created a data sample $\mathbf{x}$ using a raised sine-wave plus Gaussian noise, clipped it to the interval [0,1], then computed the feature $\mathbf{z}^*$. The original $\mathbf{x}$ was then discarded. We then computed $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$ using the method above. Figure 12 shows the results. The sample-mean of MCMC-UMS after 10000 samples matches $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$ as close as could be determined.

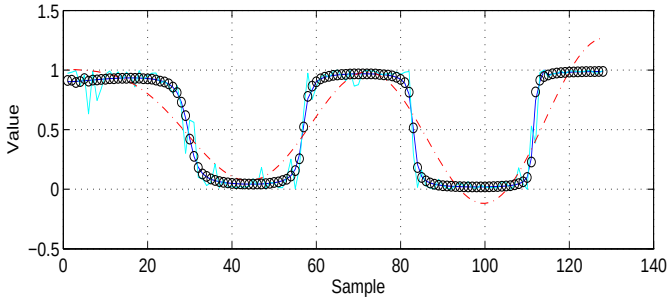

Fig. 12. The centroid estimate $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$ (circles) overlay the sample mean (dark line). One random MCMC-UMS sample is shown (light jagged line). The pseudo-inverse $\mathbf{x}_A$ (dashed curve) is seen to have values outside $(0,\ 1)$.

Fig. 13. Top left: Original cameraman image. Top right: reconstructed using 48×48 DCT coefficients. Bottom left: Maximum entropy image for doubly-bounded data, for the given 48×48 DCT coefficients. Bottom Right: Maximum entropy image for positive (singly-bounded) data, for the given 48×48 DCT coefficients - same as traditional MaxEnt method.

E. Image Example

1) Cameraman image: Let $\mathbf{x}$ be a $n^2 \times 1$ vector created from a square $n \times n$ image. The matrix $\mathbf{A}$ is constructed in order to produce the 2-dimensional DCT from the input image $\mathbf{x}$ which is of dimension $N = n^2$. For the input image, we used the "cameraman" image, down-sampled to 128×128, shown in Figure 13 (top left). For the 128×128 image, matrix $\mathbf{A}$ and the orthogonal complement matrix $\mathbf{B}$ are huge, 16384×16384 taken together. Luckily, they do not need to be explicitly constructed. Rather, products such as $\mathbf{A}'\mathbf{x}$ or $\mathbf{B}'\mathbf{x}$ and the derivatives (29) may be computed using the 2D DCT and the optimization (29) can be accomplished without second derivatives.

Figure 13 (top right) shows the same image reproduced by inverse DCT of the lower 48×48 DCT coefficients. This image is the pseudo-inverse solution (21) and has negative values and

values greater than 1, so is not a valid image.

Figure 13 (bottom left) shows the MaxEnt solution $\hat{\boldsymbol{\lambda}}(\mathbf{z}^*)$. Like the image on the top right, it is feature reproducing, but has values in $(0, 1)$. Note the better image characteristics at edges and lines.

For comparison purposes, we re-ran the maximization using the positive assumption of section IV, obtaining the solution to the maximization of (28) subject to (26), which is the classical MaxEnt image reconstruction approach, and related to a MaxEnt 2-D spectral estimate because the 2-D DCT of the image can be considered the auto-correlation function. It is the formulation used by a significant amount of the literature in image reconstruction [23]. Figure 13 (bottom right) shows the resulting image. Although still showing improvement in sharpness over Figure 13 (top right), we see the effect of having no upper bound on the pixel intensity and increased Gibbs-effect. The result is less pleasing to the eye than Figure 13 (bottom left).

VI. Sampling Application

In this section, we provide an example of the utility of sampling in a typical classification experiment.

A. Data description

We used the USPS handwritten optical character recognition (OCR) data set [33]. The data set includes 7291 training samples and 2007 testing samples of the 10 hand-written characters 0-9. Each sample is of dimension $N = 256$ (a $16\times$ 16 image) with pixel values between 0 and 1.

To limit the required processing, we limited the experiment to the three digits 3,8,9 ($M = 3$) and used only 100 randomly-chosen training samples of each digit. There were 509 samples in the testing set for the three characters. The remaining training data (that was not chosen for training) was used as a likelihood validation subset. The separate testing subset was used to measure classification performance, and we averaged the errors over ten random trials, each time selecting 100 randomly-chosen training samples of each class.

B. Benchmark Classifiers

To provide a performance benchmark, we used the following classifiers:

1) A multi-layer perceptron (ML) neural network with one hidden layer of seven nodes, which provided an average of 19.2 errors (3.77%).
2) Support vector machine (SVM). We used the SVM Light toolbox [34] to create M binary classifier functions that discriminated between the given class and the remaining classes. Optimum classification performance was observed with polynomial kernel, providing an average of 19.4 errors (3.81%).
3) Generative classifier using Gaussian mixture model (GMM). Optimum performance was observed with 24 mixture components and diagonal covariance matrices. We added a diagonal loading constant of $0.1\hat{\sigma}^2$ to the diagonal covariance elements where $\hat{\sigma}^2$ is the sample variance for the given feature. GMM provided an average of 44.3 errors (8.70%).
4) Mixture of truncated exponential distribution (TED). This is the same as the GMM approach, but the kernels are given by (43). Because the region of support of the TED distribution matches the data, we expect better performance than the GMM. Optimum classification performance was observed with 32 mixture components. TED provided an average of 41.5 errors (8.15%).
5) ILF-GMM. To better match the data to the Gaussian kernels of the GMM, we expanded the region of support to the real line using a modified inverse logistic function (ILF). Details of the ILF transformation are given in Section VIII-B. Best performance was obtained with 12 mixture components and diagonal covariance matrices with diagonal loading constant of 0.25 $\hat{\sigma}^2$. ILF-GMM provided an average of 29.7 errors (5.83%).
6) ILF-PCA-GMM. We performed PCA analysis after ILF pre-processing, projecting the data onto the top D singular vectors, then used a GMM to model the PDF of the features. PCA analysis was carried out using training data from all classes. Best performance was obtained at 3 mixture components and 48-dimensional PCA space. ILF-PCA-GMM provided an average of 22.4 errors (4.40%).

Notice that the discriminative classifiers out-perform the generative ones. Nevertheless, the generative ones in the order shown above: GMM, TED, ILF-GMM, ILF-PCA-GMM, provide decreasing error rate. We would like to see if we can continue this trend, and possibly improve upon the discriminative classifiers if we use class-dependent PCA.

C. Proposed Classifier

Our classifier was created from class-dependent linear projection operators. Learning from previous work (See [35], Section V.E), we created the generative model for a given class using all the available feature transformations. Let there be M data classes. For each class $1 \leq m \leq M$, we gathered all the training samples for class m into a $N \times n$ matrix $\mathbf{X}$, removed the mean (separately on each dimension), then performed PCA by obtaining the top $D - 1$ eigenvectors of the matrix $\mathbf{X}\mathbf{X}'$. To this, we appended the vector of ones (which is orthogonal to the eigenvectors as a result of mean removal). The result was a $N \times D$ matrix $\mathbf{A}_m$.

Since the data is limited to the unit hypercube, the feature transformations $\mathbf{z}_m = T_m(\mathbf{x})$ belong to the doubly-bounded case (Section V). Since we also wanted to experiment with the unbounded case we mapped the pixel values to $\mathcal{R}^N$ using the ILF transformation described above, then applied the method of Section III-B. Later, we corrected the likelihood values so they were referenced to the original data. In the unbounded case, the feature transformation has a second-order energy statistic, so is of dimension $D + 1$.

Basing the classifier on a class-specific model mixture with annealing factor (See [35], Section V.E), we used a non-linear

mixture of projected PDFs

$$c(\mathbf{x}|H_m) = \left(\sum_{l=1}^{M} w_{l,m} G^*(\mathbf{x}; T_l, g_{l,m})^{1/\alpha} \right)^{\alpha}, \qquad (49)$$

where $\sum_l w_{l,m} = 1$ and $G^*(\mathbf{x}; T_l, g_{l,m})$ is the maximum entropy projected PDF using feature transformation $T_l(\mathbf{x})$ and feature prior $g_{l,m}(\mathbf{z})$, which is the estimated distribution of $T_l(\mathbf{x})$ under class assumption H_m, and approximated using a GMM. The model weights $w_{l,m}$ were set to a nearly flat distribution that slightly favored the design class, so $w_{l,m} = (1+\delta(l-m)\beta)/(\beta+M)$, where β is a parameter (we used $\beta = 0.2$). We conducted some experiments to determine suitable values of D, β, α, and number of GMM components, and whether to use unbounded or doubly-bounded assumption for $\mathbf{x}$. In Figure 14 (left) , we plot number of errors obtained on the testing set (of 509 samples) averaged over the ten random trials, as a function of annealing factor α for three cases we tried. The case $D = 31$, $\beta = 1.17$, and 2 GMM components, and unbounded assumption, produced the best results.

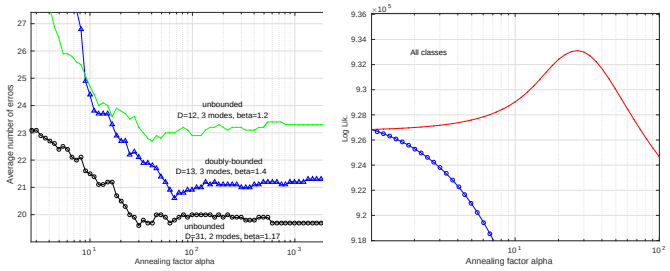

Fig. 14. Left: Average error as a function of annealing factor α. Right: Total log-likelihood (total of all classes) as a function of α with $C_m(\alpha, \beta)$ correction applied (solid lines), and with $C_m(\alpha, \beta) = 1$ (circles) for the three data classes.

Note that the classification error decreases with increasing α up to a point, reaching a minimum or flattening out. This effect of an error minimum at a particular α has been noted in various other data sets (See [35]). We would like to understand this effect better by studying it from the point of view of PDF estimation. But, to do this, we need to determine the constants $C_m(\alpha, \beta)$ by MCI, so that $c(\mathbf{x}|H_m)/C_m(\alpha, \beta)$ is a valid PDF, as explained in Section I-C, second item.

D. Sampling and Monte Carlo Integration (MCI)

As a proposal distribution $p_p(\mathbf{x}|H_m)$, we use (49) with $\alpha = 1, \beta = 0$, which simplifies to a linear mixture of projected PDFs. This provides a proposal distribution that is compatible with the integrand $c(\mathbf{x}|H_m)$. To sample from $p_p(\mathbf{x}|H_m)$, we just choose feature index l according to the uniform distribution over $[1, M]$. We then generate a sample of $\mathbf{z}_l$ from PDF $g_{l,m}(\mathbf{z}_l)$, then finally generate $\mathbf{x}$ using UMS on feature transformation $T_l(\mathbf{x})$. The constant $C_m(\alpha, \beta)$ is approximated by $C_m(\alpha, \beta) \simeq \frac{1}{K} \sum_{i=1}^{K} \frac{c(\mathbf{x}|H_m)}{p_p(\mathbf{x}|H_m)}$. The summation is taken over K samples from the proposal distribution $p_p(\mathbf{x}|H_m)$. Figure 15 (left) shows 25 examples generated in this way from the proposal distribution for digit "9". We used 10,000 samples to estimate $C_m(\alpha, \beta)$ at various values of α with β

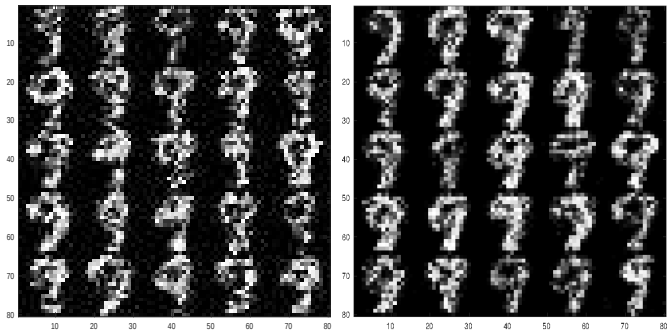

Fig. 15. Left: Twenty-five examples of digit "9" sampled by UMS from the proposal distribution $p_p(\mathbf{x}|H_m)$. Right: twenty-five samples produced by rejection samping (See Section VI-F).

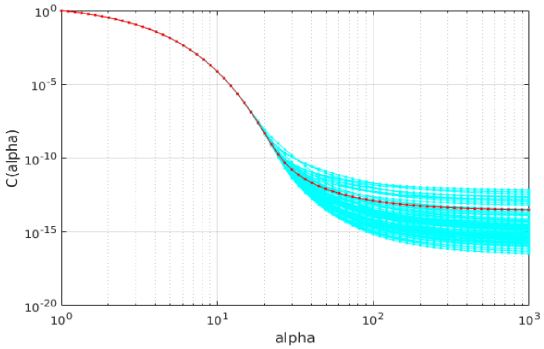

Fig. 16. Normalization constant $C_m(\alpha, \beta)$ as a function of α. 100 trials of 100 samples each are plotted along with the mean (dark line).

fixed. Figure 16 shows the results for class "9" (left). In Figure 14 (right) we see the total log-likelihood on the likelihood validation subset as a function of α for all three data classes, with $C_m(\alpha, \beta)$ correction applied and with it not applied (i.e just $c(\mathbf{x}|H_m)$). Note that without $C_m(\alpha, \beta)$, it is not possible to see the likelihood peak. The likelihood peak is in fact in agreement with the error optimum at $\alpha = 30$ in Figure 14 (left). This shows that the reason for the error minimum in Figure 14 (left) is improved PDF estimation. This new type of analysis opens the possibility of setting the parameters β, α separately for each class, for example.

E. Classifier Combination

Our classifier attained roughly the same performance as the best benchmark classifier. In an attempt to get the advantage of both classifiers, we combined them. Let $r(\mathbf{x}|H_m)$ be the output value of the SVM for class assumption H_m. The combined classifier is modeled after (2) and is written

$$h(\mathbf{x}|H_m) = c(\mathbf{x}|H_m)\, e^{a\alpha[f(3\cdot r(\mathbf{x}|H_m))-1]},$$

where a is the combining factor, α is the annealing factor defined earlier, and f is the logistic function $f(x) = 1/(1+e^{-x})$. The seemingly arbitrary processing $f(3 \cdot r(\mathbf{x}|H_m))$ approximates a posterior class probability $p(H_m|\mathbf{x})$. The combination results are shown in Figure 17 (left). Average error for the combining method had a minimum average error of 14.3

(2.8%) at a combining factor of $a = 0.7$, significantly better then the best benchmark classifier that obtained an average error of 19.2 errors (3.77%).

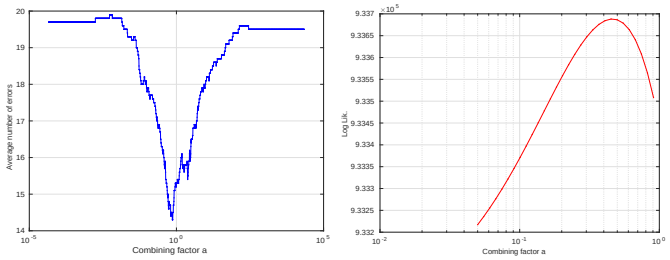

Fig. 17. Classification error (average of 10 trials) as a function of combining factor a. Left: classification error. Right: log-likelihood.

To find the cause of this error minimum, we applied MCI to estimate the normalization factor by determining the sample mean of the function $h(\mathbf{x}|H_m)/p_p(\mathbf{x}|H_m)$, over samples drawn from the proposal distribution $p_p(\mathbf{x}|H_m)$. In Figure 17 (right), we show the total likelihood value on the likelihood validation subset as a function of a, with normalization factor $C_m(\alpha, \beta)$ applied and not applied. A likelihood peak can only be seen with normalization factor applied and its location is consistent with the error minimum in Figure 17 (left). Our classifier performed significantly better than the benchmark classifiers, and the seemingly arbitrary operations that we performed could be quantitatively explained in terms of the PDF estimation.

F. Rejection Sampling

We can approximate drawing samples from the normalized distribution $h(\mathbf{x}|H_m)/C$ using rejection sampling [5]. To do this, we create a covering distribution by scaling the proposal distribution by a constant c so that it is always larger than the desired distribution $h(\mathbf{x}|H_m)$ (we can ignore the scaling). This can be approximated if we have enough samples of $p_p(\mathbf{x}|H_m)$ to be sure we have a sample near the peak of $h(\mathbf{x}|H_m)$. Let the covering distribution be $c \cdot p_p(\mathbf{x}|H_m)$. We then draw a sample $\mathbf{x}$ from $p_p(\mathbf{x}|H_m)$, then draw a uniform RV u in [0, 1]. If $u < h(\mathbf{x}|H_m)/(c \cdot p_p(\mathbf{x}|H_m))$, we accept the sample. Figure 15 (right side) shows 25 samples of class "9" created in this way. The acceptance rate was about 0.2%, which was more than adequate. Note the improved appearance of these samples compared with the proposal distribution (Left).

VII. Conclusion

We have introduced the method of uniform manifold sampling (UMS) for drawing samples from from MaxEnt PDF projection densities. We considered iterative, non-linear, and linear transformations. For linear transformations, we examined the cases of unbounded, singly-bounded, and doubly-bounded input data. We described an efficient UMS implementation using MCMC for $x_i > 0$ and for $0 \leq x_i \leq 1$, where the inversion manifold is convex and the manifold centroid (expected mean given the fixed feature value), proves to be an interesting and useful quantity. We have demonstrated the surrogate density method to compute the centroid estimate for both singly and doubly-bounded data without sampling, and used it to greatly speed up MCMC and as a generalized MaxEnt feature inversion solution that we demonstrated for MaxEnt spectral estimation and image reconstruction. We also conducted a classification experiment in which we showed the power of PDF projection and UMS to create valid generative models from hybrid classifiers combining class-dependent feature tansformations and auxiliary discriminative classifiers. We also were able to use Monte Carlo integration to predict the best parameters of the hybrid classifier, something that could lead to new methods of classifier design.

VIII. Appendix

A. Convergence of Surrogate Density to Manifold

In this appendix, we show the property that the probability mass of the surrogate density does, in fact congregate at the manifold at high dimension. As previously defined, we are given an $N \times D$ full-rank matrix $\mathbf{A}$, a fixed feature vector $\mathbf{z}^* \in \mathcal{R}^D$, and the PDF $p_s(\mathbf{x})$ with mean $\boldsymbol{\lambda}_s \in \mathcal{R}^N$ meeting the requirement $\mathbf{A}'\boldsymbol{\lambda}_s = \mathbf{z}^*$. Let the manifold be defined by $\{\mathbf{x} : \mathbf{A}'\mathbf{x} = \mathbf{z}^*\}$. As illustrated in Figure 5, we propose decomposing any vector $\mathbf{x}$ into the component $\mathbf{x}_A$ in the column space of $\mathbf{A}$ and the orthogonal component $\mathbf{x}_B$. Note that $\mathbf{x}_A = \mathbf{A}(\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{x} = \mathbf{A}(\mathbf{A}'\mathbf{A})^{-1}\mathbf{z}$. Since $\mathbf{x}_A$ depends only on $\mathbf{z}$, it is clear that any vector $\mathbf{x}$ that lies on the manifold is such that $\mathbf{x}_A$ is fixed to the value $\bar{\mathbf{x}}_A = \mathbf{A}(\mathbf{A}'\mathbf{A})^{-1}\mathbf{z}^*$. Also, as illustrated in Figure 5, let $d(\mathbf{x})$ be the distance to the manifold, $d(\mathbf{x}) = \sqrt{(\mathbf{x}_A - \bar{\mathbf{x}}_A)'(\mathbf{x}_A - \bar{\mathbf{x}}_A)}$, which may be written $d(\mathbf{x}) = \sqrt{(\mathbf{A}'\mathbf{x} - \mathbf{z}^*)'(\mathbf{A}'\mathbf{A})^{-1}(\mathbf{A}'\mathbf{x} - \mathbf{z}^*)}$ Now consider what happens if we increase the dimension of $\mathbf{x}$ by stacking n independent samples drawn from PDF $p_s(\mathbf{x})$ one on top of another so that it is dimension $n \cdot N$, and stacking matrix $\mathbf{A}$ in the same way, so that it is dimension $n \cdot N \times D$. Let these be denoted by $\mathbf{x}_n$ and $\mathbf{A}_n$, respectively. Additionally, we scale $\mathbf{A}_n$ by the factor $\frac{1}{n}$ so that the expected value of $\mathbf{A}_n'\mathbf{x}_n$ is still $\mathbf{z}^*$. This will also have the effect that $(\mathbf{A}_n'\mathbf{A}_n) = (\mathbf{A}'\mathbf{A})$. We may write $d_n(\mathbf{x}_n) = \sqrt{(\mathbf{A}_n'\mathbf{x}_n - \mathbf{z}^*)'(\mathbf{A}'\mathbf{A})^{-1}(\mathbf{A}_n'\mathbf{x}_n - \mathbf{z}^*)}$ All terms in the expression are constant except for the statistic $\mathbf{A}_n'\mathbf{x}_n$, which can be seen as an average of an increasing number of independent samples, but with known mean $\mathbf{z}^*$. By the law of large numbers, $d_n(\mathbf{x}_n)$ must go to zero as $n \to \infty$.

B. Inverse Logistic Function

To remove the effects of quantization of the USPS data, we first subtracted the value 2.42e-6 from all pixel values. Then, we added .0001 u to each pixel below 0.5, and subtracted .0001 u from each sample above 0.5, where u is a standard exponential random variable of mean 1. Then, to expand the region of support to the real line, we applied the following transformation to each pixel value x: Let $w = x(1-a)+a/2$, where $a = .00005$. Next, let $y = -\log(1/w - 1)$. The PDF of generative models computed on $\mathbf{y}$ was converted to a PDF on $\mathbf{x}$ using the rule $p(\mathbf{x}) = p(\mathbf{y}) \prod_{i=1}^{N} |\partial y_i / \partial x_i|$.

REFERENCES

[1] V. Vapnik, *The Nature of Statistical Learning*. Springer, 1999.
[2] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," in *Neural Computation 2006*, 2006.
[3] P. M. Baggenstoss, "The PDF projection theorem and the class-specific method," *IEEE Trans Signal Processing*, pp. 672–685, March 2003.
[4] P. M. Baggenstoss, "Maximum entropy pdf design using feature density constraints: Applications in signal processing," *IEEE Trans. Signal Processing*, vol. 63, June 2015.
[5] R. M. Neal, "Slice sampling," *Annals of statistics*, vol. 31, no. 3, 2003.
[6] P. M. Baggenstoss, "A modified Baum-Welch algorithm for hidden Markov models with multiple observation spaces.," *IEEE Trans. Speech and Audio*, pp. 411–416, May 2001.
[7] J. M. Lee, *Introduction to Smooth Manifolds*. Springer, 2002.
[8] N. Mitchell, M. Aanjaneya, R. Setaluri, and E. Sifakis, "Non-manifold level sets: A multivalued implicit surface representation with applications to self-collision processing," *ACM Transactions on Graphics*, vol. 36, pp. 1–9, Oct. 2015.
[9] S. M. Kay *private communication*, Apr 2013.
[10] E. T. Jaynes, "On the rationale of maximum-entropy methods," *Proceedings of IEEE*, vol. 70, no. 9, pp. 939–952, 1982.
[11] J. N. Kapur, *Maximum Entropy Models in Science and Engineering*. Wiley (Eastern), 1993.
[12] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, 2005.
[13] R. C. Aster, B. Borchers, and C. Thurber, *Parameter Estimation and Inverse Problems*. Elsevier Academic Press, 2005.
[14] S. Kay, *Fundamentals of Statisticsl Signal Processing, Estimation Theory*. Prentice Hall, Upper Saddle River, New Jersey, USA, 1993.
[15] M.-H. Chan and B. Schmeiser, "Performance of the gibbs, hit-and-run, and metropolis samplers," *Technical Report 92-18, Purdue University, Department of Statistics*, Apr. 1992.
[16] A. Y. Khincin, *Mathematical Foundations of Information Theory*. Mineaola, NY: Dover, 1957.
[17] S. Kiatsupaibul, R. Smith, and Z. Zabinsky, "An analysis of a variation of hit-and-run for uniform sampling from general regions," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 21, no. 3, 2011.
[18] R. L. Smith, "Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions," *Operations Research*, vol. 32, pp. 1296–1308, 1984.
[19] R. L. Smith, "The hit-and-run sampler: a globally reaching Markov chain sampler for generating arbitrary multivariate distributions," *Proceedings of the 1996 Winter Simulation Conference*, 1996.
[20] R. M. Neal, "Slice sampling," *The Annals of Statistics*, vol. 31, no. 3, pp. 705–767, 2003.
[21] N. A. Malik, "One and two dimensional maximum entropy spectral estimation," *MIT PhD Thesis*, Nov 1981.
[22] J. P. Burg, "The relationship between maximum entropy and maximum likelihood spectra," *Geophysics*, vol. 37, no. 2, pp. 375–376, 1971.
[23] S. J. Wernecke and L. R. D'Addario, "Maximum entropy image reconstruction," *IEEE Trans. Computers*, vol. C-26, no. 4, pp. 351–364, 1977.
[24] G. Wei and H. Zhen-Ya, "A new algorithm for maximum entropy image reconstruction," in *Proceedings of ICASSP-87*, vol. 12, pp. 595–597, April 1987.
[25] S. Kay, *Modern Spectral Estimation: Theory and Applications*. Prentice Hall, 1988.
[26] B. D. Ripley, *Stochastic Simulation*. John Wiley & Sons, 1987.
[27] P. Mermelstein, "Distance measures for speech recognition, psychological and instrumental," *Pattern Recognition and Artificial Intelligence*, p. 374388, 1976.
[28] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book, Version 3.4*. Cambridge University Engineering Department, 2006.
[29] B. Milner and X. Shao, "Speech reconstruction from mel-frequency cepstral coefficients using a source-filter model.," in *INTERSPEECH*, Citeseer, 2002.
[30] P. M. Baggenstoss, "The class-specific classifier: Avoiding the curse of dimensionality (tutorial)," *IEEE Aerospace and Electronic Systems Magazine, special Tutorial addendum*, vol. 19, pp. 37–52, January 2004.
[31] V. P. Singh, *Entropy, Theory and its Applications in Environmental and Water Engineering*. John Wiley & Sons, 2013.
[32] K. Conrad, "Probability distributions and maximum entropy," *Unpublished article*, 2013.
[33] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans PAMI*, vol. 16, no. 5, p. 550554, 1994.
[34] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods - Support Vector Learning* (B. Schölkopf, C. Burges, and A. Smola, eds.), ch. 11, pp. 169–184, Cambridge, MA: MIT Press, 1999.
[35] P. M. Baggenstoss, "Class-specific model mixtures for the classification of acoustic time-series," *IEEE Trans. AES*, Aug. 2016.

Paul M Baggenstoss. Dr. Baggenstoss received his PhD in electrical engineering (statistical signal processing) at the University of Rhode Island (URI) in 1990. From 1979 to 1996, he was with Raytheon Co, Portsmouth, RI. He joined the Naval Undersea Warfare Center (NUWC) Newport, RI, in 1996, where he applied statistical signal processing and classification theory to problems in underwater acoustics.

During 2000, he was a visiting scientist at the University of Erlangen, Erlangen, Germany, and during 2010, he was a exchange scientist at Fraunhofer FKIE, Wachtberg, Germany. In 2015, he joined Fraunhofer FKIE, where he conducts research in classification theory and machine leaning, with applications to speaker identification. He is the author of several patents and numerous conference and journal papers. He is the recipient 2002 URI Excellence Award in Science and Technology, the 2004 NAVSEA Scientist of the year award, and the 2004 NUWC Excellence in Science award.