Probabilistic Latent Variable Model for Sparse Decompositions of Non-negative Data

Madhusudana Shashanka, *Student Member, IEEE,*
Bhiksha Raj, and Paris Smaragdis, *Senior Member, IEEE*

***Abstract*— An important problem in data-analysis tasks is to find suitable representations that make hidden structure in the data explicit. In this paper, we present a probabilistic latent variable model that is equivalent to a matrix decomposition of non-negative data. Data is modeled as histograms of multiple draws from an underlying generative process. The model expresses the generative distribution as a mixture of *hidden distributions* which capture the latent structure. We extend the model to incorporate sparsity constraints by using an entropic prior. We derive algorithms for parameter estimation and show how the model can be applied for unsupervised feature extraction and supervised classification tasks.**

***Index Terms*— Probabilistic Algorithms, Latent Class Models, Sparse Decomposition, Entropic Prior, MAP estimation**

I. Introduction

Component-wise decompositions have long enjoyed ubiquitous use in machine learning problems. Popular approaches such as Principal Components Analysis (PCA), Independent Components Analysis (ICA) and Projection Pursuit are frequently employed for various tasks such as feature discovery or object extraction and their statistical properties have made them indispensable tools for machine learning applications. More recently, Non-negative Matrix Factorization (NMF; [12]) introduced a new desirable property for component decompositions, that of non-negativity. Non-negativity has proven to be a very valuable property for researchers working with positive data. Methods not using non-negativity are bound to discover a set of bases that contain negative elements, and then employ cross-cancellation between them in order to approximate the input. Such components with negative elements are hard to interpret in positive data domains and are often used for their statistical properties and not for the insight they provide. NMF, which approximates the input as additive combinations of non-negative components, has been found to provide meaningful components on a variety of data types such as images [12] and audio magnitude spectra [20]. However, NMF is not defined in a probabilistic framework. It is neither a generative model nor a discriminative model and this can pose difficulties in situations where it is used alongside other models. Also, it does not provide a way to utilize any kind of *a priori* information available about the data.

Another desirable property for component-wise decompositions that has spawned an active area of research is the concept of sparsity. It has its roots in the field of sensory coding, where the idea of efficient coding has been proposed as a way to extract the intrinsic structure in sensory signals. The goal is to find a set of basis vectors that span the input space such that only a few of the basis vectors are required to describe a particular input vector. In other words, only a small number of mixture weights with which these basis vectors combine to explain the input vector have significant magnitudes while all the rest have zero or negligible magnitudes. Olshausen and Field [15], who use sparse coding to explain natural image statistics, obtain such a code by trying to reduce the entropies of these mixture weights. Field [5] provides a detailed theoretical treatment of sparse coding, especially in the case of *overcomplete* codes.

M. Shashanka is with the Hearing Research Center, Boston University, Boston, MA 02215, USA. Email: shashanka@cns.bu.edu.

B. Raj and P. Smaragdis are with Mitsubishi Electric Research Labs, Cambridge, MA 02139, USA. Email: {bhiksha, paris}@merl.com.

Non-negativity and sparsity together are desirable properties to have for basis decomposition techniques, but a statistical underpinning for such techniques is just as important. There are two main contributions of this paper. We first present a generative latent variable model that provides a statistical framework in which one can extract non-negative bases from data. Instead of modeling the data directly, the model works on the underlying generative distribution and expresses it as a mixture of latent distributions which can be interpreted as basis vectors. The mixture weights with which the basis vectors combine are also modeled as distributions, thus implicitly guaranteeing non-negativity in the whole model. We then propose algorithms to impose sparsity in this statistical framework.

Rest of the paper is organized as follows. We present the latent variable model in Section II and describe algorithms for learning the parameters. In Section III, we show how sparsity can be incorporated in this framework. In section IV, we try to visualize and understand the workings of the model by using a geometric interpretation. We show how the model relates to NMF and latent class models in Section V. Section VI describes experiments where we show how the model can be used for unsupervised feature extraction and supervised classification. Finally, we end the paper with conclusions in Section VII.

II. Latent Variable Model

In this section, we present the latent variable model, provide equations for parameter estimation and then show that it is equivalent to a basis decomposition in the probability domain.

Consider a random process characterized by the probability $P(f)$ of drawing a feature unit f in a given draw. Let the random variable f take values from the set $\{1, 2, \ldots, F\}$. Let us assume that $P(f)$ is unknown and what one can observe instead is the result of multiple draws from the process. In other words, we observe feature *counts* i.e. the number of times feature f is observed after repeated draws. We can approximate the *generative distribution* $P(f)$ by using the normalized set of counts.

Now suppose we also know that $P(f)$ is comprised of R *hidden distributions* or *latent factors*. The observation in a given draw might come from any one of the R distributions. The distributions are selected according to a probability distribution that remains constant across draws in a given experiment. We are allowed to run multiple experiments and observe feature counts for each experiment. The probabilities according to which distributions are selected vary from experiment to experiment. Our task is to characterize these hidden distributions.

Let us represent the hidden distributions by $P(f|z)$ - the probability of observing feature f conditioned on a *latent variable* z. The probability of picking the z-th distribution in the n-th experiment can be represented by $P_n(z)$. We can now formally write the model as

$$P_n(f) = \sum_z P(f|z) P_n(z), \tag{1}$$

where $P_n(f)$ gives the overall probability of observing feature f in the n-th experiment. Here, the multinomial distributions

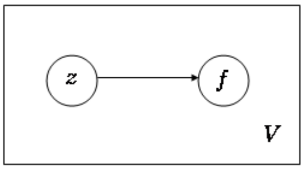


Fig. 1. Graphical model for the random process underlying the generation of a data vector. Circles represent variables, the surrounding box represents repeated draws and the arrow represents dependence. z is the hidden variable, f is the feature drawn, and V is the total number of draws.

$P(f|z)$ can be thought of as *basis vectors* that are characteristic to all experiments. $P_n(z)$ are mixture weights that signify the contribution of $P(f|z)$ towards $P_n(f)$. The subscript n indicates that mixture weights change from experiment to experiment.

The random process generating counts in the n-th experiment can be summarized as

1) Pick a latent variable z with probability $P_n(z)$.
2) Pick feature f from the multinomial distribution $P(f|z)$.
3) Repeat the above two steps V times,

where V is total number of draws in the experiment. Figure 1 shows the graphical model depicting the process.

Let V_{fn} represent the feature count of feature f in the n-th experiment. Given these feature counts, one can analyze and estimate the parameters of the model as we show below. Notice that any process that generates counts or histograms is suitable for analysis by the model. If we have non-negative data, we can model them as histograms generated from underlying distributions. Each data vector can be modeled as the result of a separate random experiment. Thus, we can use the model for analysis of non-negative data.

A. Parameter Estimation

Given feature counts V_{fn}, we aim to estimate $P(f|z)$ and $P_n(z)$. The components are randomly initialized and re-estimated through iterations as given below, which are derived using the Expectation Maximization (EM) algorithm. The Expectation (E) step can be written as

$$P_n(z|f) = \frac{P_n(z)P(f|z)}{\sum_z P_n(z)P(f|z)}, \tag{2}$$

and the Maximization (M) step can be written as

$$P(f|z) = \frac{\sum_n V_{fn}P_n(z|f)}{\sum_f \sum_n V_{fn}P_n(z|f)}, P_n(z) = \frac{\sum_f V_{fn}P_n(z|f)}{\sum_z \sum_f V_{fn}P_n(z|f)}. \tag{3}$$

The E-step and M-step equations are alternated until a termination condition is met. Detailed derivation is shown in supplemental material. The EM algorithm guarantees that the above multiplicative updates converge to a local optimum.

B. Latent Variable Model as Basis Decomposition

We can write the model given by equation (1) in matrix form as $\mathbf{p}_n = \mathbf{W}\mathbf{h}_n$, where $\mathbf{p}_n$ is a column vector indicating $P_n(f)$, $\mathbf{h}_n$ is a column vector indicating $P_n(z)$, and $\mathbf{W}$ is the $F \times R$ matrix with the (f, z)-th element corresponding to $P(f|z)$. Concatenating all column vectors $\mathbf{p}_n$ and $\mathbf{h}_n$ as matrices $\mathbf{P}$ and $\mathbf{H}$ respectively, one can write the model as

$$\mathbf{P} = \mathbf{W}\mathbf{H}. \tag{4}$$

This formulation is similar to matrix decompositions such as PCA, ICA and NMF. We have additional constraints that the columns of $\mathbf{P}$, $\mathbf{W}$ and $\mathbf{H}$, being probability distributions, should be positive and sum to unity. Thus, the model is equivalent to a matrix decomposition which operates in the probability distribution space. Non-negativity is implicitly guaranteed since we are dealing with probabilities. We want to point out that the latent variable model can be generalized where the random process generates a multidimensional feature $\mathbf{f}$ in a given draw. In that case, the above matrix decomposition generalizes to a tensor decomposition but we shall not explore it further in this paper.

We can write the update equations (2) and (3) also in matrix form. Writing the normalization steps separately, we have

$$\begin{aligned} W_{fr} &= W_{fr} \sum_n \frac{H_{rn}V_{fn}}{(WH)_{fn}}, \quad W_{fr} = \frac{W_{fr}}{\sum_f W_{fr}}, \qquad \text{and,} \\ H_{rn} &= H_{rn} \sum_f \frac{W_{fr}V_{fn}}{(WH)_{fn}}, \quad H_{rn} = \frac{H_{rn}}{\sum_r H_{rn}}, \end{aligned} \tag{5}$$

where A_{ij} represents the ij-th entry of matrix $\mathbf{A}$. The above equations are similar to NMF update equations as we shall point out in Section V.

III. Sparsity in the Latent Variable Model

Sparse coding refers to a representational scheme where, of a set of components that may be combined to compose data, only a small number are combined to represent any particular input. In this section, we present a brief motivation for this concept of *sparsity* and show how one can incorporate it in the framework of the latent variable model.

The idea of sparsity originated from attempts at understanding the general information processing strategy employed by biological sensory systems. The assumption is that the goal of sensory coding is to transform the input in such a manner that reduces the redundancy present among the elements of the input stream. Consider the context of basis decomposition techniques. The data vector $\mathbf{v}$ (or the underlying generative distribution in the case of a latent variable model) is approximated as $\mathbf{Wh}$ where the columns of $\mathbf{W}$ are basis vectors and elements h_i of the vector $\mathbf{h}$ are the mixture weights. In this context, this goal of *efficient coding* is equivalent to finding a set of basis vectors that forms a complete code (i.e. spans the input space) and results in the mixture weights being as statistically independent as possible over an ensemble of inputs. One way of achieving this, as suggested by Field [5], is to have a representational scheme where only a few (out of a large population) of the basis vectors are required to explain any particular data vector. As Olshausen and Field [15] explain, the existence of any statistical dependencies among a set of variables h_i may be discerned whenever the the joint entropy is less than the sum of the individual entropies, i.e. $\mathcal{H}(h_1, h_2, \ldots, h_r) < \sum_i \mathcal{H}(h_i)$, where $\mathcal{H}$ is the entropy. They explain that a possible strategy for reducing statistical dependencies is to lower the individual entropies $\mathcal{H}(h_i)$. Thus, reducing entropies of mixture weights is equivalent to having a sparse code of basis vectors.

Different metrics have been proposed to measure sparsity. These metrics are used as constraints during parameter estimation of the model so that sparsity is enforced. They correspond to different cost functions that penalize the objective function during estimation. Consider a distribution $\boldsymbol{\theta}$ on which sparsity is desired. Some approaches use variants of the L_p norm of $\boldsymbol{\theta}$ as the cost function (eg. [9]) while other approaches use various approximations of entropy of $\boldsymbol{\theta}$ as the cost function (eg. [15]). In this paper, instead of using approximations for entropy we use entropy itself as a sparsity metric and seek to directly reduce it during estimation.

We use the concept of *entropic prior* which has been used in the *maximum entropy* literature (see [10], [19]) to enforce sparsity. Given a probability distribution $\boldsymbol{\theta}$, the entropic prior is defined as

$$P_e(\boldsymbol{\theta}) \propto e^{-\alpha \mathcal{H}(\boldsymbol{\theta})}, \tag{6}$$

where $\mathcal{H}(\boldsymbol{\theta}) = -\sum_i \theta_i \log \theta_i$ is the entropy of the distribution and α is a weighting factor. Positive values of α favor distributions with lower entropies while negative values of α favor distributions with higher entropies. Imposing this prior with positive α during *maximum a posteriori* estimation is a way to minimize entropy which will result in a sparse $\boldsymbol{\theta}$ distribution. The distribution $\boldsymbol{\theta}$ could correspond to the basis functions $P(f|z)$ or the mixture weights $P_n(z)$ or both.

We use the EM algorithm to derive the update equations. Let us examine the case where both $P(f|z)$ and $P_n(z)$ have the entropic prior. The set of parameters to be estimated is given by $\Lambda = \{P(f|z), P_n(z)\}$. The *a priori* distribution over the parameters, $P(\Lambda)$, corresponds to the entropic priors. We can write $\log P(\Lambda)$, the log- prior, as

$$\alpha \sum_z \sum_f P(f|z) \log P(f|z) + \beta \sum_n \sum_z P_n(z) \log P_n(z), \tag{7}$$

where α and β are parameters[1] indicating the degree of sparsity desired in $P(f|z)$ and $P_n(z)$ respectively. As before, we can write the E-step as

$$P_n(z|f) = \frac{P_n(z)P(f|z)}{\sum_z P_n(z)P(f|z)}. \tag{8}$$

The M-step reduces to the equations

$$\frac{\sum_n V_{fn} P_n(z|f)}{P(f|z)} + \alpha + \alpha \log P(f|z) + \rho_z = 0, \tag{9}$$

$$\frac{\sum_f V_{fn} P_n(z|f)}{P_n(z)} + \beta + \beta \log P_n(z) + \tau_n = 0, \tag{10}$$

where ρ_z and τ_n are Lagrange multipliers. The above M-step equations are systems of simultaneous transcendental equations for $P(f|z)$ and $P_n(z)$. Brand [1] proposes a method to solve such equations using the Lambert $\mathcal{W}$ function (see [3] for details about this function). It can be shown that $P(f|z)$ and $P_n(z)$ can be estimated as

$$\hat{P}(f|z) = \frac{-\xi/\alpha}{\mathcal{W}(-\xi e^{1+\rho_z/\alpha}/\alpha)}, \tag{11}$$

$$\hat{P}_n(z) = \frac{-\omega/\beta}{\mathcal{W}(-\omega e^{1+\tau_n/\beta}/\beta)}, \tag{12}$$

where we have let ξ represent $\sum_n V_{fn} P_n(z|f)$ and ω represent $\sum_f V_{fn} P_n(z|f)$. Equations (9) and (11) (and the equation pair (10) and (12)) form a set of fixed-point iterations that typically converge in 2-5 iterations (see [1]). Brand [2] provides details on computing the Lambert's $\mathcal{W}$ function.

[1] α and β can also take negative values, in which case the entropies of corresponding distributions are increased during estimation.

The final update equations are given by equation (8), and the fixed-point equation-pairs (9), (11) and (10), (12). Details of the derivation are provided in supplemental material.

Notice that the above update equations reduce to the maximum likelihood updates of equations (2) and (3) when α and β are set to zero. In most applications, sparsity is usually desired on either $P(f|z)$ or $P_n(z)$ and not on both simultaneously. Sparse coding or efficient coding corresponds to imposing sparsity on the mixture weights $P_n(z)$. Sparsity on the basis vectors $P(f|z)$ can be useful in certain feature extraction applications as will be shown by an example in Section VI.

IV. Geometry of the Latent Variable Model

The latent variable model as given by eqn. (1) expresses an F-dimensional distribution $P_n(f)$ as a mixture of R F-dimensional basis distributions $P(f|z)$. Being probability distributions, they are points in the $(F-1)$-dimensional simplex. In case of 3-dimensional distributions (a 3-dimensional input space), the generative distributions and basis vectors lie within the *Standard 2-Simplex* (the plane defined by points on each axis which are unit distance from the origin) and hence are easy to visualize.

To understand and visualize the workings of the model, we created an artificial data set of 400 3-dimensional distributions. We want to emphasize at this point that the input to the model is always a histogram of multiple draws from an underlying generative distribution. Every point in the data set we generated actually corresponds to a normalized histogram. In other words, every point corresponds to the result of a different experiment. In this section, we use the terms "data points" and "experiments" interchangeably.

We applied the latent variable model on the artificial dataset. The model expresses the generative distribution for every data point as a linear combination of the basis vectors where the mixture weights are positive and sum to unity. Geometrically, this implies that a given generative distribution is expressed as a point within the *convex-hull* formed by the basis vectors. This is illustrated in Figure 2 for 2 and 3 basis vectors. In most applications of basis decompositions such as NMF or PCA, the number of basis vectors extracted is far less than the dimensionality of the input space. In such a *compact code* where dimensionality of representation is reduced, the goal is to represent all the likely inputs with a relatively small number of vectors with minimal loss in the description of the input space. The left panel of Figure 2 with 2 basis vectors corresponds to a *compact code* while the right panel with 3 basis vectors corresponds to a *complete code*. One can also choose to have an *overcomplete code* with more basis vectors than the input dimensionality and imposing sparsity on the mixture weights in this case can give desirable properties as we show below.

A. Effect of Number of Basis Vectors

The number of basis vectors used in a latent variable model significantly affects the performance of the model. Figure 2 shows that when the number of basis functions is increased from 2 (corresponding to a *compact code*) to 3 (corresponding to a complete code), the resulting approximation improves from a line to a plane. All points which lie outside the line can now be

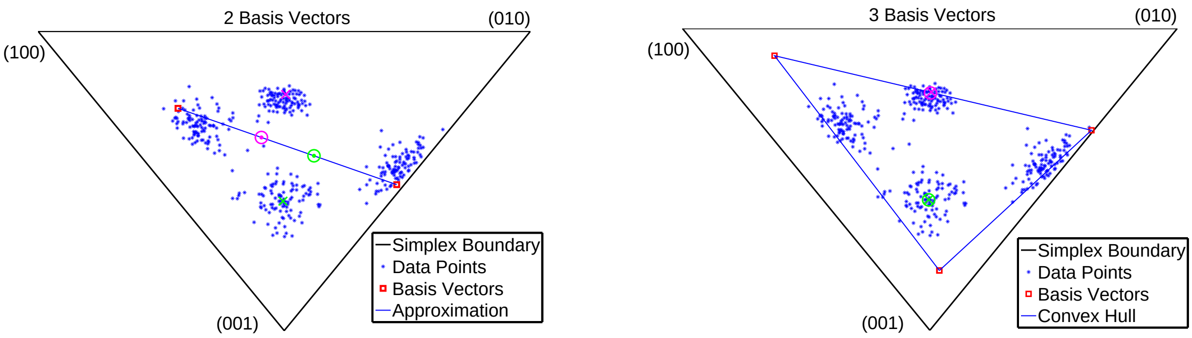


Fig. 2. Illustration of the latent variable model on 3-dimensional distributions. Both panels show distributions represented within the *Standard 2-Simplex* given by $\{(001), (010), (100)\}$. 2 Basis Vectors (Left) and 3 Basis Vectors (Right) extracted from 400 data points are shown. The model approximates data vectors as points lying on the line approximation (left) or within the convex hull (right) formed by the basis vectors. Also shown are two data points (marked by the magenta and green crosses) and their approximations by the model (shown by the circles). As one can see, the model gets more accurate as the number of basis vectors increases from a *compact code* of 2 basis vectors to a *complete code* of 3 basis vectors.

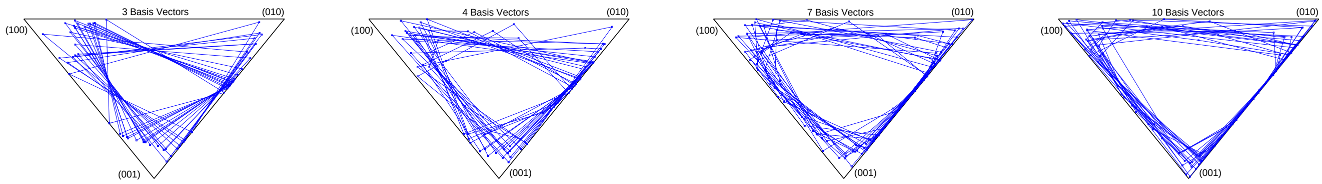


Fig. 3. Illustration of the effect of number of basis vectors on the latent variable model applied on 3-dimensional distributions. Points are represented within the *Standard 2-Simplex* given by $\{(001), (010), (100)\}$. The model was applied on the data set of 400 points shown in Figure 2 to extract 3, 4, 7, and 10 basis vectors. Each case consisted of 20 repeated runs and the resulting convex hulls formed by the basis vectors were plotted as shown in the panels from left to right. Notice that increasing the number of basis vectors enlarges the sizes of convex hulls.

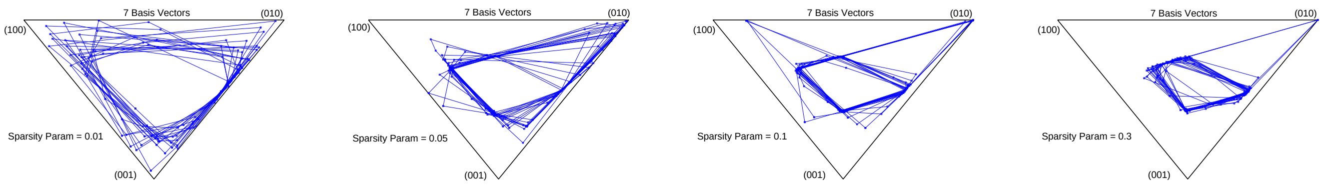
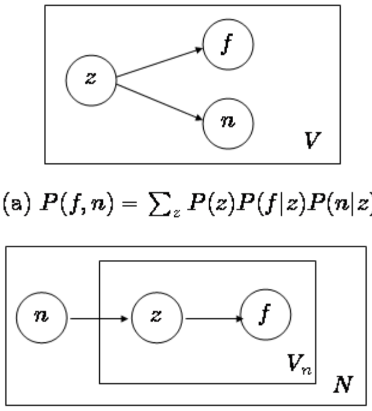


Fig. 4. Illustration of the effect of sparsity on the latent variable model applied on 3-dimensional distributions. Points are represented within the *Standard 2-Simplex* given by $\{(001), (010), (100)\}$. The latent variable model was applied on data shown in Figure 2 to extract 7 basis vectors with different values of the sparsity parameter on the mixture weights. There were 20 repeated runs for a given value of the sparsity parameter and the resulting convex hulls are plotted as shown. Increasing the sparsity of mixture weights makes the resulting convex hulls more compact.

accurately represented by 3 basis vectors since they lie *within* the convex hull. However, as one increases the number of basis functions beyond the input dimensionality, the resulting convex hulls "expand" around the data as shown in Figure 3. This larger set of basis vectors can accurately represent the data but are less characteristic of the distribution of data points. In other words, the new set of basis vectors is less informative about the data set. Consider the extreme case where we have the set of corners of the 2-simplex as basis vectors. They accurately represent the data set but do not provide any information. This is because they can represent not just this dataset but *any* other data set with perfect accuracy.

B. *Effect of Sparsity*

In the latent variable model, sparsity is imposed on a particular parameter by reducing its entropy. It is easy to understand the effects of sparsifying basis vectors. As the entropy decreases, they are pushed towards the corners of the input simplex. In the extreme case where entropy of each basis vector is reduced to zero, the basis vectors are given by the corners of the simplex. The effect of imposing sparsity on the mixture weights to get a *sparse code* of basis vectors is more interesting. Figure 4 shows that as the sparsity parameter is increased, convex hulls formed by the basis vectors get more "compact" around the data. This corresponds to a *sparse-overcomplete* code comprising a large number of basis vectors but few of them contributing towards explaining any particular data point. This happens when the basis vectors themselves are more data-like, or in other words, holistic representations of the input space. The idea of sparse-overcomplete latent variable model has recently been used in audio source separation tasks to obtain better performance [18]. In Section VI, we show an example where it results in improved classification performance.



(a) $P(f, n) = \sum_z P(z)P(f|z)P(n|z)$

(b) $P(f, n) = P(n) \sum_z P(f|z)P(z|n)$

Fig. 5. Graphical models for two-dimensional latent class model. Circles represent variables, a box surrounding them indicates how many times they should be drawn and arrows indicate dependence. (a) z represents the hidden variable, f and n are the features drawn in the two dimensions in a given draw, and V is the total number of draws. (b) z represents the hidden variable, f is the feature drawn in a given draw, V_n is the total number draws for the n-th data vector, and N is the total number of data vectors.

C. *Information encoding in the model*

Let us examine how the model encodes information about the dataset. Information about the dataset is encoded by both the basis vectors and the mixture weights. Global characteristics are encoded by basis vectors while local characteristics (of individual data points) are encoded by mixture weights. Basis vectors correspond to characteristics of the random process that remained invariant during all the experiments while mixture weights correspond to characteristics specific to the experiments. Increasing the number of basis vectors in an overcomplete code means that the basis vectors become less informative about the data set. The information is pushed from basis vectors to mixture weights. In the extreme case where the corners of the *2-Simplex* correspond to the basis vectors, all the information about the dataset is encoded by the mixture weights and the basis vectors themselves provide no information. On the other hand, as one increases sparsity of mixture weights, basis vectors become more data-like. Information about the data set is pushed from the mixture weights to the basis vectors. In the extreme sparse-overcomplete case where each data-point is itself a basis vector, all information about the data set is coded by the basis vectors and mixture weights provide no information.

V. Relation to Latent Class Models and NMF

The latent variable model we have presented is conceptually related to *Latent Class Models* and numerically similar to Non-negative Matrix Factorization. In this section, we describe how the model relates to these techniques.

A. *Latent Class Models*

The model presented in Section II is a variation of a *Latent Class Model*. Latent class models have been used in the field of social and behavioral sciences as an analysis method. The models enable one to attribute the observationss as being due to latent factors (eg. [6], [7], [11]). The main characteristic of these models is *conditional independence* - multivariate data are modeled as belonging to latent classes such that random variables within a class are independent of one another. In its general form, a latent class model expresses a K-dimensional distribution as a mixture where each component of the mixture is a product of one-dimensional marginal distributions. Mathematically, it can be written as

$$P(\mathbf{x}) = \sum_z P(z) \prod_{j=1}^{K} P(x_j|z), \tag{13}$$

where $P(\mathbf{x})$ is a K-dimensional distribution of the random variable $\mathbf{x} = x_1, x_2, \ldots, x_K$. Mixture components are indexed by the latent variable z and $P(x_j|z)$ are one-dimensional marginal distributions. For two-dimensional data in the form of the $F \times N$ matrix $\mathbf{V}$, the model can be expressed as

$$P(f, n) = \sum_z P(z)P(f|z)P(n|z), \text{ or} \tag{14}$$

$$\mathbf{P} = \mathbf{WSH} \tag{15}$$

in matrix form, where $F \times N$ matrix $\mathbf{P}$ represents the two-dimensional distribution $P(f, n)$, $\mathbf{W}$ is an $F \times R$ matrix with the f-th entry of the z-th column representing $P(f|z)$, $\mathbf{S}$ is an $R \times R$ diagonal matrix where the z-th diagonal element represents $P(z)$, and $\mathbf{H}$ is an $R \times N$ matrix with the n-th element of the z-th row representing $P(n|z)$. Random variables corresponding to both dimensions are considered as features and are treated symmetrically. This is depicted by the graphical model in Figure 5(a). Convolutive extensions of this model were recently proposed by [21] and have been applied to various acoustic processing tasks [22].

In our case, we have N data vectors of dimension F and we do not wish to treat both dimensions symmetrically. One can use a different factorization[2] as follows:

$$P(f, n) = P(n) \sum_z P(f|z)P(z|n), \text{ or} \tag{16}$$

$$\mathbf{P} = \mathbf{WHS} \tag{17}$$

in matrix form, where $\mathbf{P}$ represents the two-dimensional distribution $P(f, n)$, $\mathbf{W}$ is an $F \times R$ matrix with the f-th entry of the z-th column representing $P(f|z)$, $\mathbf{H}$ is an $R \times N$ matrix with the z-th entry of the n-th column representing $P(z|n)$, and $\mathbf{S}$ is an $N \times N$ diagonal matrix with the n-th diagonal element equal to $P(n)$. Figure 5(b) shows the graphical model for this factorization. Hofmann [8], motivated by applications in semantic analysis of text corpora, introduced this model as Probabilistic Latent Semantic Analysis.

The model presented in this paper does not explicitly estimate $P(n)$. It was proposed by Raj and Smaragdis [17] in the context of separating speakers from single-channel acoustic recordings. We consider each data vector $\mathbf{v}_n$ independently and model N one-dimensional distributions $P_n(f)$ instead of the two-dimensional distribution $P(f, n)$. This is equivalent to using the latent class model of equation (14) on every data vector independently. Treating the two dimensions differently helps in clear interpretation of the resulting decomposition as basis vectors and mixture weights.

[2]Instead, we can use $P(f, n) = P(f) \sum_z P(n|z)P(z|f)$ (or in matrix form: $\mathbf{P}_{F\times N} = \mathbf{S}_{F\times F}\mathbf{W}_{F\times R}\mathbf{H}_{R\times N}$, where subscripts denote matrix sizes and $\mathbf{S}$ is a diagonal matrix), where basis vectors are over dimension n and given by rows of $\mathbf{H}$. This is numerically equivalent to using equation (16) or (17) with the input dimensions transposed.

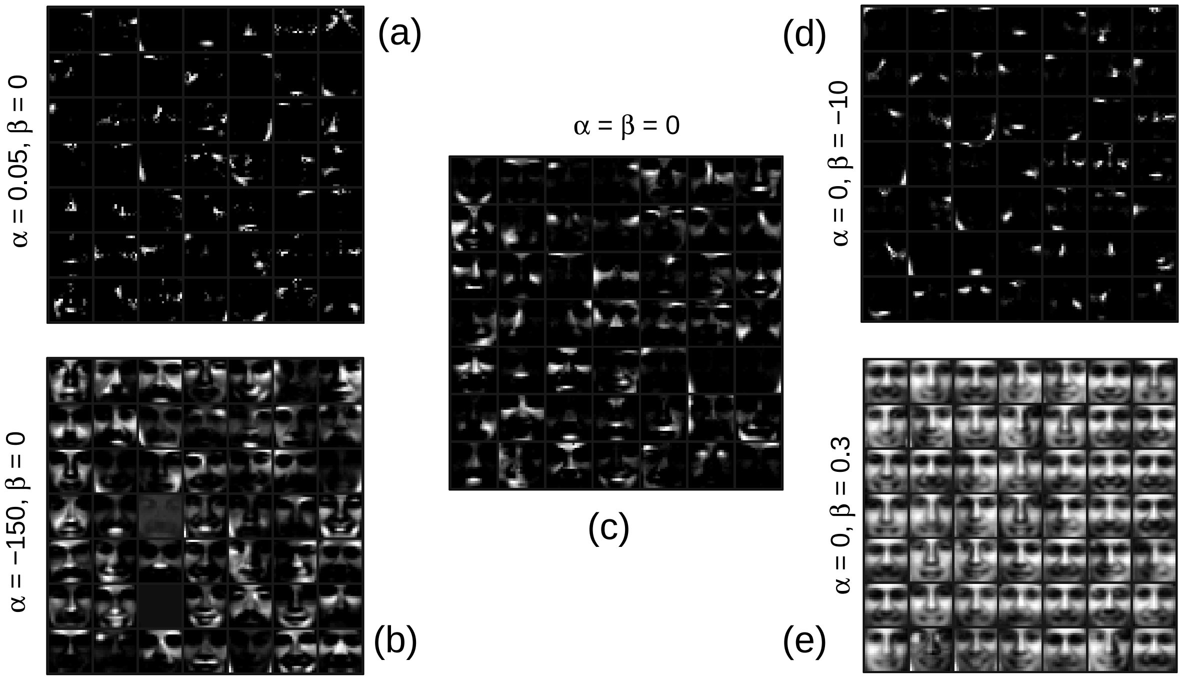


Fig. 6. Basis images extracted from the *CBCL Database* (http://cbcl.mit.edu/software-datasets/FaceData2.html) using the latent variable model. Panel (c) shows 49 basis images extracted without using sparsity. These are qualitatively similar to the basis vectors obtained by NMF (not shown). Notice that they are not entirely parts-like representations. Panels (a) and (b) show results of varying α - the sparsity parameter on the basis vectors. Panels (d) and (e) show the effects of varying β - the sparsity parameter on mixture weights. Parts-like representations are obtained when one imposes sparsity on the basis vectors (a) or increases entropy of the mixture weights (d). Increasing entropy of basis vectors (b) and decreasing entropy of the mixture weights (e) leads to holistic face-like representations.

B. Non-negative Matrix Factorization

Non-negative Matrix Factorization was introduced by [12] to find non-negative parts-based representation of data. Given a $F \times N$ matrix $\mathbf{V}$ where each column corresponds to a data vector, NMF approximates it as a product of non-negative matrices $\mathbf{W}$ and $\mathbf{H}$, i.e. $\mathbf{V} \approx \mathbf{WH}$, where $\mathbf{W}$ is a $F \times R$ matrix and $\mathbf{H}$ is a $R \times N$ matrix. The columns of $\mathbf{W}$ can be thought of as *basis vectors* that are optimized for the linear approximation of $\mathbf{V}$.

The optimal choice of matrices $\mathbf{W}$ and $\mathbf{H}$ are defined by those non-negative matrices that minimize the reconstruction error between $\mathbf{V}$ and $\mathbf{WH}$. Different error functions have been proposed which lead to different update rules (eg. [12], [13]). Shown below are multiplicative update rules derived by [13] using an error measure similar to the Kullback-Leibler divergence:

$$\begin{aligned} H_{rn} &\leftarrow H_{rn} \frac{\sum_f W_{fr} V_{fn}/(WH)_{fn}}{\sum_f W_{fr}}, \\ W_{fr} &\leftarrow W_{fr} \frac{\sum_n H_{rn} V_{fn}/(WH)_{fn}}{\sum_n H_{rn}}, \end{aligned} \tag{18}$$

where A_{ij} represents the value at i-th row and the j-th column of matrix $\mathbf{A}$. One can see that the EM update equations for the latent variable model given by equations (5) are similar to the above NMF updates. They differ in the normalization factors.

Several authors have noted the shortcomings of standard NMF and suggested extensions to incorporate sparsity. Most approaches use variants of a penalty function on $\mathbf{H}$ during estimation to enforce sparsity on $\mathbf{H}$. The penalty functions include the L_1 norm [4], [14], a combination of L_1 or L_2 norms [9] or other functions of $\mathbf{H}$ [4]. Some approaches such as [16] use a modified approximation that enforces sparsity. Convergence properties of some of the approaches are unknown ([4], [14]). We have proposed a method with probabilistic foundations using entropy as the sparsity measure. The method we have proposed has good convergence properties as guaranteed by the EM algorithm. In addition, our method can be generalized to have multidimensional features which is equivalent to a tensor decomposition.

VI. Experiments

In this section, we describe results of applying the latent variable model on real life data and show how it can be used for feature extraction and classification tasks.

A. Feature Extraction

Lee and Seung [12] applied NMF on the *CBCL database* of faces and showed that the basis functions extracted had localized features that fit well with intuitive notions of parts of faces. We applied the latent variable model on the database and Figure 6(c) shows the results. Equations (2) and (3) were used to update parameters and data was preprocessed as was done in the original study[3]. Bases extracted by the model are qualitatively similar to those extracted from NMF (see [9], [12]).

[3]The *CBCL database* consists of 2429 frontal view face images, hand-aligned in a 19×19 grid. Following [12], the grayscale intensities were first linearly scaled so that the pixel mean and standard deviation were equal to 0.25, and then clipped to the range [0, 1].

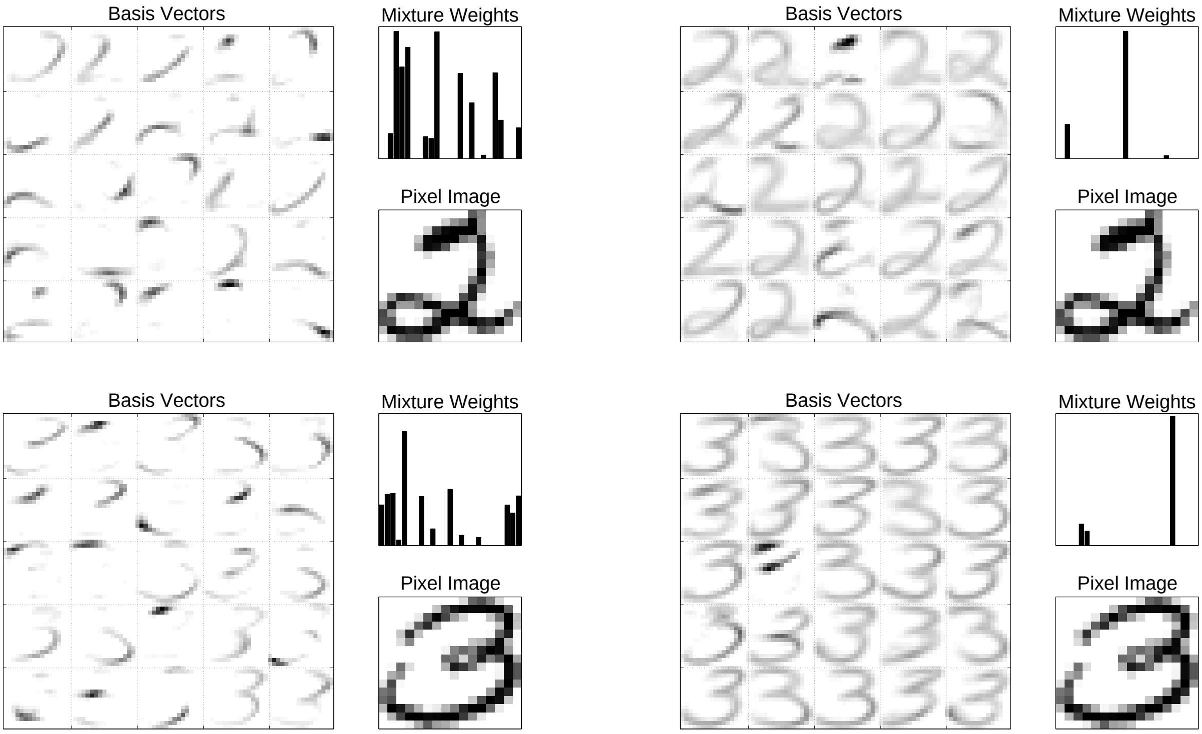


Fig. 7. 25 Basis images extracted for class “2” (Top Panels) and class “3” (Bottom panels) from training data without sparsity on mixture weights (Left Panels, sparsity parameter = 0) and with sparsity on mixture weights (Right Panels, sparsity parameter = 0.2). Basis images combine in proportion to the mixture weights shown to result in the pixel images shown.

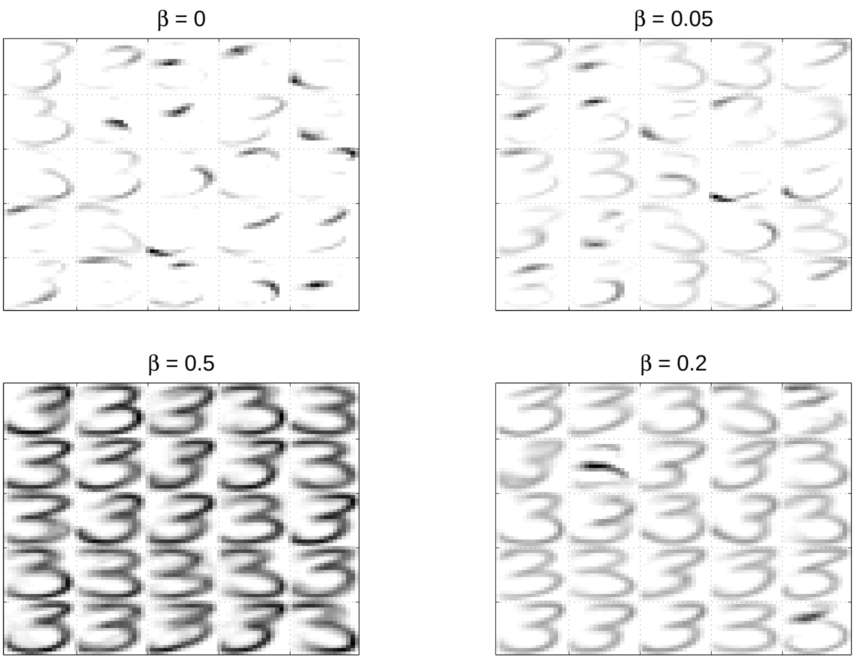


Fig. 8. 25 basis functions learned from training data for class “3” with increasing sparsity parameters on the mixture weights. The sparsity parameter was set to (from top-left in clockwise direction) 0, 0.05, 0.2 and 0.5 respectively. Unlike the basis vectors of Figure 6, increasing the sparsity parameter of mixture weights produces basis vectors which are holistic representations of the input space instead of parts-like features.

However, the extracted bases are not entirely parts based representations as can be seen from the figure. Notice that compared to holistic representations, parts-based representations should have lower entropy. We ran experiments on the *CBCL Database* by applying sparsity on the basis vectors to see if it resulted in parts-based representations. Results are shown in Figure 6(a). Decreasing the entropy of basis vectors leads to parts-like representations. Qualitatively similar results can be obtained by *increasing* the entropy of mixture weights as shown in Figure 6(d).

Instead of parts-like representations, one can obtain holistic representations by imposing sparsity on the mixture weights as shown by Figure 6(e). Qualitatively similar results can be obtained by increasing the entropy of basis vectors as shown in Figure 6(b).

B. Classification

We now provide a simple example of handwritten digit classification using the latent variable framework and show how sparsity applied on the mixture weights affects classification. We used the USPS Handwritten Digits database (from http://www.cs.toronto.edu/∼roweis/data.html) which has 1100 examples for each digit class. We randomly chose 100 examples from each class and separated them as the test set. The remaining examples were used for training.

Training and testing procedures were as follows. During training, separate sets of basis vectors were learned for each class. Figure 7 shows 25 bases images extracted for the digits "2" and "3" respectively. During testing, basis vectors $P^k(f|z)$ were fixed and mixture weights $P^k(z)$ were estimated to obtain mixture distribution $P^k(f) = \sum_z P^k(f|z)P^k(z)$, where the superscript k indicates the class label. For a given test data vector $\mathbf{v}$, this process was repeated with basis vectors from each class and the likelihood $\mathcal{L}^k = \sum_f v_f \log P^k(f)$ was computed. The vector $\mathbf{v}$ was assigned to the class for which likelihood was the highest.

Results are shown in Figure 9. As one can see, imposing sparsity improves classification performance in almost all cases. Figure 8 shows four sets of basis vectors learned for class "3" with different sparsity values on the mixture weights. As the sparsity parameter is increased, basis vectors tend to be holistic representations of the input space. This is consistent with improved classification performance - as the representation of basis vectors get more holistic, the more *unlike* they become when compared to bases of other classes. Thus, there is a lesser chance that basis vectors of one class can combine to approximate an input vector in another class, thereby improving performance.

VII. Conclusions

In this paper, we presented a probabilistic generative model that enforces non-negativity implicitly. We showed that it is equivalent to a basis decomposition in the probability space. We showed how the model could be extended to incorporate sparsity by adding an entropic prior during estimation. We presented a geometric view that helped us visualize the workings of the model and the effects of sparsity on its performance. We clarified how the model relates to NMF and latent class models. We experimentally verified the applicability of proposed models for unsupervised feature extraction tasks as well as supervised classification tasks. Future research directions include extensions of this work to model time-varying data (such as speech spectrograms) and evaluating other suitable priors that can better capture the structure present in data.

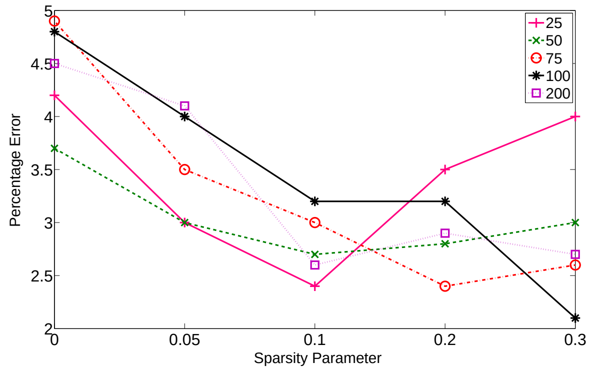


Fig. 9. Results of the classification experiment. The legend shows number of basis vectors used. Notice that imposing sparsity almost always leads to better classification performance. In the case of 100 basis vectors, error rate comes down by almost 50% when a sparsity parameter of 0.3 is imposed.

References

[1] ME Brand. Pattern discovery via entropy minimization. In *Uncertainty 99: AISTATS 99*, 1999.

[2] ME Brand. Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation*, 1999.

[3] RM Corless, GH Gonnet, DEG Hare, DJ Jeffrey, and DE Knuth. On the lambert $\mathcal{W}$ function. *Advances in Computational mathematics*, 1996.

[4] J Eggert and E Korner. Sparse coding and nmf. *Neural Networks*, 2004.

[5] DJ Field. What is the goal of sensory coding? *Neural Computation*, 1994.

[6] LA Goodman. Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika*, 61:215–231, 1974.

[7] BF Green Jr. Latent structure analysis and its relation to factor analysis. *Journal of the American Statistical Association*, 47:71–76, 1952.

[8] T Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42:177–196, 2001.

[9] PO Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5, 2004.

[10] ET Jaynes. *Papers on probability, statistics and statistical mechanics*. Kluwer Academic, 1982.

[11] PF Lazarfeld and NW Henry. *Latent Structure Analysis*. Boston: Houghton Mifflin, 1968.

[12] DD Lee and HS Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401, 1999.

[13] DD Lee and HS Seung. Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, 13, 2001.

[14] M Morup and MN Schmidt. Sparse non-negative matrix factor 2-d deconvolution. Technical report, Technical University of Denmark, 2006.

[15] BA Olshausen and DJ Field. Emergence of simple-cell properties by learning a sparse code for natural images. *Nature*, 381, 1996.

[16] A Pascaul-Montano, JM Carazo, K Kochi, D Lehmann, and RD Pascaul-Marqui. Nonsmooth nonnegative matrix factorization. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 28(3), 2006.

[17] B Raj and P Smaragdis. Latent variable decomposition of spectrograms for single channel speaker separation. In *IEEE WASPAA*, 2005.

[18] MVS Shashanka, B Raj, and P Smaragdis. Sparse overcomplete decomposition for single channel speaker separation. In *ICASSP*, 2007.

[19] J Skilling. Classic maximum entropy. In J Skilling, editor, *Maximum Entropy and Bayesian Methods*. Kluwer Academic, 1989.

[20] P Smaragdis and JC Brown. Non-negative matrix factorization for polyphonic music transcription. In *IEEE WASPAA*, 2003.

[21] P Smaragdis and B Raj. Shift-invariant probabilistic latent component analysis. *Journal of Machine Learning Research (submitted)*, 2007.

[22] P Smaragdis, B Raj, and MVS Shashanka. A probabilistic latent variable model for acoustic modeling. In *NIPS Workshop on Advances in Modeling for Acoustic Processing*, 2006.