Probabilistic Latent Variable Models as Non-Negative Factorizations

Madhusudana Shashanka, Bhiksha Raj, Paris Smaragdis

***Abstract*— In this paper, we present a family of probabilistic latent variable models that can be used for analysis of non-negative data. We show that there are strong ties between non-negative matrix factorization and this family, and provide some straightforward extensions which can help in dealing with shift-invariances, higher order decompositions and sparsity constraints. We argue through these extensions that the use of this approach allows for rapid development of complex statistical models for analyzing non-negative data.**

***Index Terms*— Non-Negative Matrix Factorization, Latent Variable Models**

I. Introduction

Techniques to analyze non-negative data are required in several applications such as analysis of images, text corpora and audio spectra to name a few. A variety of techniques have been proposed for the analysis of such data, such as non-negative PCA [1], non-negative ICA [2], non-negative matrix factorization (NMF) [3] etc. The goal of all of these techniques is to explain the given non-negative data as a guaranteed non-negative linear combination of a set of non-negative "bases" that represent realistic "building blocks" for the data. Of these, probably the most developed is non-negative matrix factorization, with much recent research devoted to the topic [4], [5], [6]. All of these approaches view each data vector as a point in an N-dimensional space and attempt to identify the bases that best explain the distribution of the data within this space. For the sake of clarity, we will refer to data that represent vectors in any space as *point* data.

A somewhat related, but separate topic that has garnered much research over the years is the analysis of histograms of multi-variate data. Histogram data represent the counts of occurrences of a set of events in a given data set. The aim here is to identify the statistical factors that affect the occurrence of data through the analysis of these counts and appropriate modeling of the distributions underlying them. Such analysis is often required in the analysis of text, behavioral patterns etc. A variety of techniques, such as probabilistic latent semantic analysis [7], latent Dirichlet allocation [8], etc. and their derivatives have lately become quite popular. Most, if not all of them can be related to a class of probabilistic models, known in the behavioral sciences community at *Latent Class Models* [9], [10], [11], that attempt to explain the observed histograms as having been drawn from a set of latent classes, each with its own distribution. For clarity, we will refer to histograms and collections of histograms as *histogram* data.

In this paper, we argue that techniques meant for analysis of histogram data can be equally effectively employed for decomposition of non-negative point data as well, by interpreting the latter as scaled histograms rather than vectors. Specifically, we show that the algorithms used for estimating the parameters of a latent class model are numerically equivalent to the update rules for one form of NMF. We also propose alternate latent variable models for histogram decomposition that are similar to those commonly employed in the analysis of text, to decompose point data and show that these too are identical to the update rules for NMF. We will generically refer to the application of histogram-decomposition techniques to point data as probabilistic decompositions[1].

Beyond simple equivalences to NMF, the probabilistic decomposition approach has several advantages, as we explain. Non-negative PCA/ICA and NMF are primarily intended for matrix-like two-dimensional characterizations of data – the analysis is obtained for matrices that are formed by laying data vectors side-by-side. They do not naturally extend to higher-dimensional tensorial representations, this has been often accomplished by implicit unwrapping the tensors into a matrix. However, the probabilistic decomposition naturally extends from matrices to tensors of arbitrary dimensions.

It is often desired to control the form or structure of the learned bases and their projections. Since the procedure for learning the bases that represent the data is statistical, probabilistic decomposition affords control over the form of the learned bases through the imposition of *a priori* probabilities, as we will show. Constraints such as sparsity can also be incorporated through these priors.

We also describe extensions to the basic probabilistic decomposition framework that permits shift-invariance along one or more of the dimensions (of the data tensor) that can abstract convolutively combined bases from the data.

The rest of the paper is organised as follows. Since, the probabilistic decomposition approach we promote in this paper is most analogous to Non-negative Matrix Factorization (NMF) among all techniques that analyse non-negative point data, we begin with a brief discussion of NMF. We present the family of latent variable models in Section III that we will employ for probabilistic decompositions. We present tensor generalizations in Section IV-A and convolutive factorizations in Section IV-B. In Section IV-C we discuss extensions such as incorporation of sparsity and in Section IV-D we present aspects of geometric interpretation of these decompositions.

[1] This must not be confused with approaches that model the distribution of the set of vectors. In our approach the vectors themselves are histograms, or, alternately, scaled probability distributions.

II. Non-Negative Matrix Factorization

Non-negative Matrix Factorization was introduced by [3] to find non-negative parts-based representation of data. Given an $M \times N$ 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 $M \times K$ matrix and $\mathbf{H}$ is a $K \times N$ matrix. The above approximation can be written column by column as $\mathbf{v}_n \approx \mathbf{W}\mathbf{h}_n$, where $\mathbf{v}_n$ and $\mathbf{h}_n$ are the n-th columns of $\mathbf{V}$ and $\mathbf{H}$ respectively. In other words, each data vector $\mathbf{v}_n$ is approximated by a linear combination of the columns of $\mathbf{W}$, weighted by the entries of $\mathbf{h}_n$. The columns of $\mathbf{W}$ can be thought of as *basis vectors* that, when combined with appropriate *mixture weights* (entries of the columns of $\mathbf{H}$), provide a 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], [3]). Shown below are multiplicative update rules derived by [3] using an error measure similar to the Kullback-Leibler divergence:

$$\begin{aligned} W_{mk} &\leftarrow W_{mk}\sum_n \frac{V_{mn}}{(WH)_{mn}} H_{kn}, \quad W_{mk} \leftarrow \frac{W_{mk}}{\sum_m W_{mk}}, \\ H_{kn} &\leftarrow H_{kn}\sum_m W_{mk}\frac{V_{mn}}{(WH)_{mn}}, \qquad (1) \end{aligned}$$

where A_{ij} represents the value at i-th row and the j-th column of matrix $\mathbf{A}$.

III. Latent Variable Models

In its simplest form, NMF expresses an $M \times N$ data matrix $\mathbf{V}$ as the product of non-negative matrices $\mathbf{W}$ and $\mathbf{H}$. The idea is to express the data vectors (columns of $\mathbf{V}$) as a combination of a set of *basis components* or *latent factors* (columns of $\mathbf{W}$). Below, we show that a class of probabilistic models employing latent variables, known in the field of social and behavioral sciences as *Latent Class Models* (eg., [11], [9], [13]), are equivalent to NMF.

Let us represent the two dimensions of the matrix $\mathbf{V}$ by x_1 and x_2 respectively. We can consider the non-negative entries $V_{x_1x_2}$ as having been generated by an underlying probability distribution $P(x_1, x_2)$. Variables x_1 and x_2 are multinomial random variables where x_1 can take one out of a set of M values in a given draw and x_2 can take one out of a set of N values in a given draw. In other words, one can model V_{mn}, the entry in row m and column n, as the number of times features $x_1 = m$ and $x_2 = n$ were picked in a set of repeated draws from the distribution $P(x_1, x_2)$. Unlike NMF which tries to characterize the observed data directly, latent class models characterize the underlying distribution $P(x_1, x_2)$. This subtle difference of interpretation preserves all the advantages of NMF, while overcoming some of its limitations by providing a framework that is easy to generalize, extend and interpret.

There are two ways of modeling $P(x_1, x_2)$ and we consider them separately below.

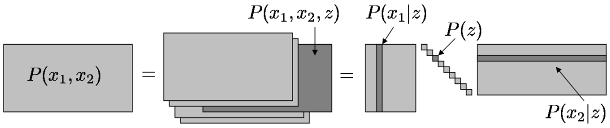

Fig. 1. Latent variable model of equation (2) as matrix factorization.

A. Symmetric Factorization

Latent class models enable one to attribute the observations as being due to hidden or latent factors. The main characteristic of these models is conditional independence - multivariate data are modeled as belonging to latent classes such that the random variables within a latent class are independent of one another. The model expresses a multivariate distribution such as $P(x_1, x_2)$ as a mixture where each component of the mixture is a product of one-dimensional marginal distributions. In the case of two dimensional data such as $\mathbf{V}$, the model can be written mathematically as

$$P(x_1, x_2) = \sum_{z \in \{1,2,\ldots,K\}} P(z)P(x_1|z)P(x_2|z). \qquad (2)$$

In the above equation, z is a latent variable that indexes the hidden components and takes values from the set $\{1, \ldots, K\}$. This equation assumes the *principle of local independence*, whereby the latent variable z renders the observed variables x_1 and x_2 independent. This model was presented independently as *Probabilistic Latent Component Analysis* (PLCA) by [14]. The aim of the model is to characterize the distribution underlying the data as shown above by learning the parameters so that hidden structure present in the data becomes explicit.

The model can be expressed as a matrix factorization. Representing the parameters $P(x_1|z)$, $P(x_2|z)$ and $P(z)$ as entries of matrices $\mathbf{W}$, $\mathbf{G}$ and $\mathbf{S}$ respectively where

- $\mathbf{W}$ is a $M \times K$ matrix such that W_{mk} corresponds to the probability $P(x_1 = m|z = k)$,
- $\mathbf{G}$ is a $K \times N$ matrix such that G_{kn} corresponds to the probability $P(x_2 = n|z = k)$, and
- $\mathbf{S}$ is a $K \times K$ diagonal matrix such that S_{kk} corresponds to the probability $P(z = k)$,

one can write the model of equation (2) in matrix form as

$$\begin{aligned} \mathbf{P} &= \mathbf{WSG}, \text{or equivalently,} \qquad (3) \\ \mathbf{P} &= \mathbf{WH}, \qquad (4) \end{aligned}$$

where the entries of matrix $\mathbf{P}$ correspond to $P(x_1, x_2)$ and $\mathbf{H} = \mathbf{SG}$. Figure 1 illustrates the model schematically.

Parameters can be estimated using EM algorithm. The update equations for the parameters can be written as

$$\begin{aligned} P(z|x_1, x_2) &= \frac{P(z)P(x_1|z)P(x_2|z)}{\sum_z P(z)P(x_1|z)P(x_2|z)}, \\ P(x_i|z) &= \frac{\sum_{j\in\{1,2\},j\neq i} V_{x_1x_2}P(z|x_1,x_2)}{\sum_{x_1,x_2} V_{x_1x_2}P(z|x_1,x_2)}, \\ P(z) &= \frac{\sum_{x_1,x_2} V_{x_1x_2}P(z|x_1,x_2)}{\sum_{z,x_1,x_2} V_{x_1x_2}P(z|x_1,x_2)}. \qquad (5) \end{aligned}$$

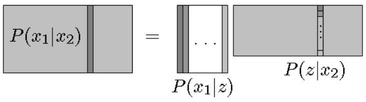

Fig. 2. Latent variable model of equation (7) as matrix factorization.

Writing the above update equations in matrix form using $\mathbf{W}$ and $\mathbf{H}$ from equation (3), we obtain

$$\begin{aligned} W_{mk} &\leftarrow W_{mk} \sum_n \frac{V_{mn}}{(WH)_{mn}} H_{kn}, \quad W_{mk} \leftarrow \frac{W_{mk}}{\sum_m W_{mk}}, \\ H_{kn} &\leftarrow H_{kn} \sum_m W_{mk} \frac{V_{mn}}{(WH)_{mn}}, \quad H_{kn} \leftarrow \frac{H_{kn}}{\sum_{k,n} H_{kn}} \end{aligned} \quad (6)$$

The above equations are identical to the NMF update equations of equation (1) upto a scaling factor in $\mathbf{H}$. This is due to the fact that the probabilistic model decomposes $\mathbf{P}$ which is equivalent to a normalized version of the data $\mathbf{V}$. [14] presents detailed derivation of the update algorithms and comparison with NMF update equations. This model has been used in analyzing image and audio data among other applications (eg., [14], [15], [16]).

B. Asymmetric Factorization

The latent class model of equation (2) considers each dimension symmetrically for factorization. The two dimensional distribution $P(x_1, x_2)$ is expressed as a mixture of two-dimensional latent factors where each factor is a product of one-dimensional marginal distributions. Now, consider the following factorization of $P(x_1, x_2)$:

$$\begin{aligned} P(x_1, x_2) &= P(x_i) P(x_j|x_i), \\ P(x_j|x_i) &= \sum_z P(x_j|z) P(z|x_i), \end{aligned} \quad (7)$$

where $i, j \in \{1, 2\}$, $i \neq j$ and z is a latent variable. This version of the model with asymmetric factorization is popularly known as *Probabilistic Latent Semantic Analysis* (PLSA) in the topic-modeling literature [7].

Without loss of generality, let $j = 1$ and $i = 2$. We can write the above model in matrix form as $\mathbf{q}_n = \mathbf{W}\mathbf{g}_n$, where $\mathbf{q}_n$ is a column vector indicating $P(x_1|x_2)$, $\mathbf{g}_n$ is a column vector indicating $P(z|x_2)$, and $\mathbf{W}$ is a matrix with the (m, k)-th element corresponding to $P(x_1 = m|z = k)$. If z takes K values, $\mathbf{W}$ is a $M \times K$ matrix. Concatenating all column vectors $\mathbf{q}_n$ and $\mathbf{g}_n$ as matrices $\mathbf{Q}$ and $\mathbf{G}$ respectively, one can write the model as

$$\begin{aligned} \mathbf{Q} &= \mathbf{WG}, \text{or equivalently} \\ \mathbf{V} &= \mathbf{WGS} = \mathbf{WH}, \end{aligned} \quad (8)$$

where $\mathbf{S}$ is a $N \times N$ diagonal matrix whose n-th diagonal element is the sum of the entries of $\mathbf{v}_n$ (the n-th column of $\mathbf{V}$), and $\mathbf{H} = \mathbf{GS}$. Figure 2 provides a schematic illustration of the model.

Given data matrix $\mathbf{V}$, parameters $P(x_1|z)$ and $P(z|x_2)$ are estimated by iterations of equations derived using the EM algorithm:

$$\begin{aligned} P(z|x_1, x_2) &= \frac{P(z|x_2) P(x_1|z)}{\sum_z P(z|x_2) P(x_1|z)} \\ P(x_1|z) &= \frac{\sum_{x_2} V_{x_1 x_2} P(z|x_1, x_2)}{\sum_{x_1, x_2} V_{x_1 x_2} P(z|x_1, x_2)} \\ P(z|x_2) &= \frac{\sum_{x_1} V_{x_1 x_2} P(z|x_1, x_2)}{\sum_{x_1} V_{x_1 x_2}}. \end{aligned} \quad (9)$$

Writing the above equations in matrix form using $\mathbf{W}$ and $\mathbf{H}$ from equation (8), we obtain

$$\begin{aligned} W_{mk} &\leftarrow W_{mk} \sum_n \frac{V_{mn}}{(WH)_{mn}} H_{kn}, \quad W_{mk} \leftarrow \frac{W_{mk}}{\sum_m W_{mk}}, \\ H_{kn} &\leftarrow H_{kn} \sum_m W_{mk} \frac{V_{mn}}{(WH)_{mn}}. \end{aligned} \quad (10)$$

The above set of equations is exactly identical to the NMF update equations of equation (1). See [17], [18] for detailed derivation of the update equations. The equivalence between NMF and PLSA has also been pointed out by [19]. The model has been used for the analysis of audio spectra (eg., [20]), images (eg., [17], [21]) and text corpora (eg., [7]).

IV. Model Extensions

The popularity of NMF comes mainly from its empirical success in finding "useful components" from the data. As pointed out by several researchers, NMF has certain important limitations despite the success. We have presented probabilistic models that are numerically closely related to or identical to one of the widely used NMF update algorithms. Despite the numerical equivalence, the methodological difference in approaches is important. In this section, we outline some advantages of using this alternate probabilistic view of NMF.

The first and most straightforward implication of using a probabilistic approach is that it provides a theoretical basis for the technique. And more importantly, the probabilistic underpinning enables one to utilize all the tools and machinery of statistical inference for estimation. This is crucial for extensions and generalizations of the method. Beyond these obvious advantages, below we discuss some specific examples where utilizing this approach is more useful.

A. Tensorial Factorization

NMF was introduced to analyze two-dimensional data. However, there are several domains with non-negative multi-dimensional data where a multi-dimensional correlate of NMF could be very useful. This problem has been termed as Non-negative Tensor Factorization (NTF). Several extensions of NMF have been proposed to handle multi-dimensional data (eg., [22], [6], [4], [5]). Typically, these methods flatten the tensor into a matrix representation and proceed further with analysis. Conceptually, NTF is a natural generalization of NMF but the estimation algorithms for learning the parameters, however, do not lend themselves to extensions easily. Several issues contribute to this difficulty. We do not present the reasons here due to lack of space but a detailed discussion can be found in [6].

Now, consider the symmetric factorization case of the latent variable model presented in Section III-A. This model is naturally suited for generalizations to multiple dimensions. In its general form, the model expresses a K-dimensional distribution as a mixture, where each K-dimensional component of the mixture is a product of one-dimensional marginal distributions. Mathematically, it can be written as

$$P(\mathbf{x}) = \sum_z P(z) \prod_{j=1}^{K} P(x_j|z), \tag{11}$$

where $P(\mathbf{x})$ is a K-dimensional distribution of the random variable $\mathbf{x} = x_1, x_2, \ldots, x_K$. z is the latent variable indexing the mixture components and $P(x_j|z)$ are one-dimensional marginal distributions. Parameters are estimated by iterations of equations derived using the EM algorithm and they are:

$$\begin{aligned} R(\mathbf{x}, z) &= \frac{P(z)\prod_{j=1}^{N} P(x_j|z)}{\sum_{z'} P(z')\prod_{j=1}^{N} P(x_j|z')} && (12) \\ P(z) &= \sum_j \sum_{x_j} P(\mathbf{x}) R(\mathbf{x}, z) && (13) \\ P(x_j|z) &= \frac{\sum_{i:i\neq j} \sum_{x_i} P(\mathbf{x}) R(\mathbf{x}, z)}{P(z)} && (14) \end{aligned}$$

In the two-dimensional case, the update equations reduce to equations (5).

To illustrate the kind of output of this algorithm consider the following toy example. The input $P(\mathbf{x})$ was the 3 dimensional distribution shown in the upper left plot in figure 3. This distribution can also be seen as a rank 3 positive tensor. It is clearly composed out of two components, each being an isotropic Gaussian with means at $\mu_1 = 11, 11, 9$ and $\mu_2 = 14, 14, 16$ and variances $\sigma_1^2 = 1$ and $\sigma_2^2 = 1/2$ respectively. The bottom row of plots show the derived sets of $P(x_j|z)$ using the estimation procedure we just described. We can see that each of them is composed out of a Gaussian at the expected position and with the expected variance. The approximated $P(\mathbf{x})$ using this mode is shown in the top right. Other examples of applications on more complex data and a detailed derivation of the algorithm can be found in [23], [14].

B. Convolutive Decompositions

Given a two-dimensional dataset, NMF finds hidden structure along one dimension (column-wise) that is characteristic to the entire dataset. Consider a scenario where there is localized structure present along both dimensions (rows and columns) that has to be extracted from the data. An example dataset would be an acoustic spectrogram of human speech which has structure along both frequency and time. Traditional NMF is unable to find structure across both dimensions and several extensions have been proposed to handle such datasets (eg., [24], [25]).

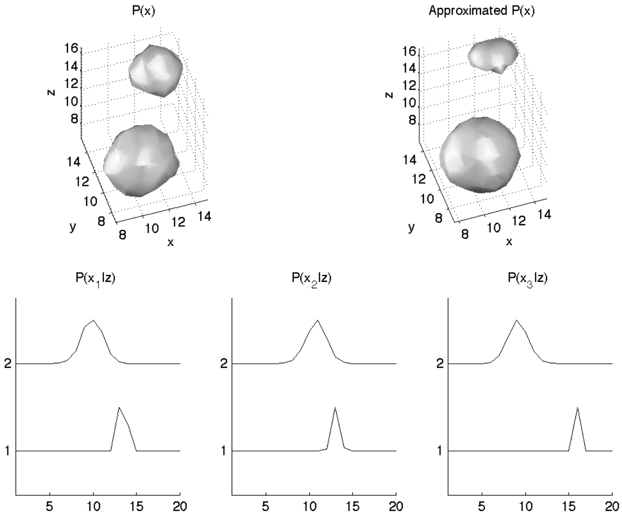

Fig. 3. An example of a higher dimensional positive data decomposition. An isosurface of the original input is shown at the top left, the approximation by the model in eq. 11 is shown in the top right and the extracted marginals (or factors) are shown in the lower plots.

The latent variable model can be extended for such datasets and the parameter estimation still follows a simple EM algorithm based on the principle of maximum likelihood. The model, known as a *shift-invariant* version of PLCA, can be mathematically written as [23]

$$P(\mathbf{x}) = \sum_z \left(P(z) \int P(\mathbf{w}, \boldsymbol{\tau}|z) P(\mathbf{h} - \boldsymbol{\tau}|z) d\boldsymbol{\tau} \right) \tag{15}$$

where the *kernel distribution* $P(\mathbf{w}, \boldsymbol{\tau}|z) = 0, \forall \boldsymbol{\tau} \notin \mathcal{R}$ where $\mathcal{R}$ defines a local convex region along the dimensions of $\mathbf{x}$. Similar to the simple model of equation (2), the model expresses $P(\mathbf{x})$ as a mixture of latent components. But instead of each component being a simple product of one-dimensional distributions, the components are convolutions between a multi-dimensional "kernel distribution" and a multi-dimensional "impulse distribution". The update equations for the parameters are:

$$R(\mathbf{x}, \boldsymbol{\tau}, z) = \frac{P(z) P(\mathbf{w}, \boldsymbol{\tau}|z) P(\mathbf{h} - \boldsymbol{\tau}|z)}{\sum_{z'} P(z') \int P(\mathbf{w}, \boldsymbol{\tau}'|z') P(\mathbf{h} - \boldsymbol{\tau}'|z') d\boldsymbol{\tau}'} \tag{16}$$

$$P(z) = \int R(\mathbf{x}, z) d\mathbf{x} \tag{17}$$

$$P(\mathbf{w}, \boldsymbol{\tau}|z) = \frac{\int P(\mathbf{x}) R(\mathbf{x}, \boldsymbol{\tau}, z) d\mathbf{h}}{P(z)} \tag{18}$$

$$P(\mathbf{h}|z) = \frac{\int P(\mathbf{w}, \mathbf{h} + \boldsymbol{\tau}) R(\mathbf{w}, \mathbf{h} + \boldsymbol{\tau}, \boldsymbol{\tau}, z) d\mathbf{w} d\boldsymbol{\tau}}{\int P(\mathbf{w}, \mathbf{h}' + \boldsymbol{\tau}) R(\mathbf{w}, \mathbf{h}' + \boldsymbol{\tau}, \boldsymbol{\tau}, z) d\mathbf{h}' d\mathbf{w} d\boldsymbol{\tau}} \tag{19}$$

Detailed derivation of the algorithm can be found in [14]. The above model is able to deal with tensorial data just as well as matrix data. To illustrate this model, consider the picture in the top left of figure 4. This particular image is a rank-3 tensor (x, y, color). We wish to discover the underlying components that make up this image. The components are the digits 1, 2, 3 and appear in various spatial locations, thereby necessitating a "shift-invariant" approach. Using the aforementioned algorithm we obtain the results shown in

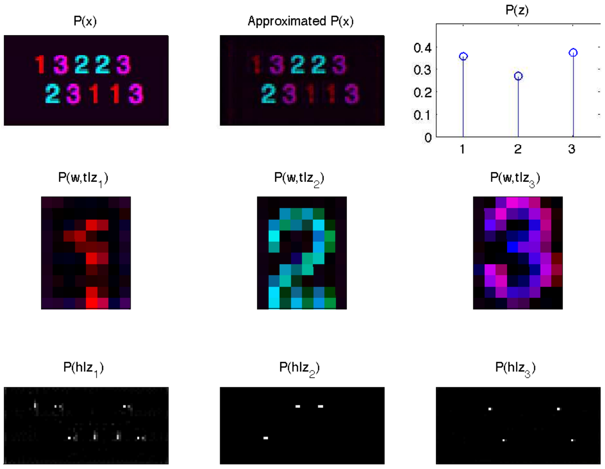

Fig. 4. An example of a higher dimensional shift-invariant positive data decomposition. The original input is shown at the top left, the approximation by the model in eq. 11 is shown in the top middle and the extracted kernels and impulses are shown in the lower plots.

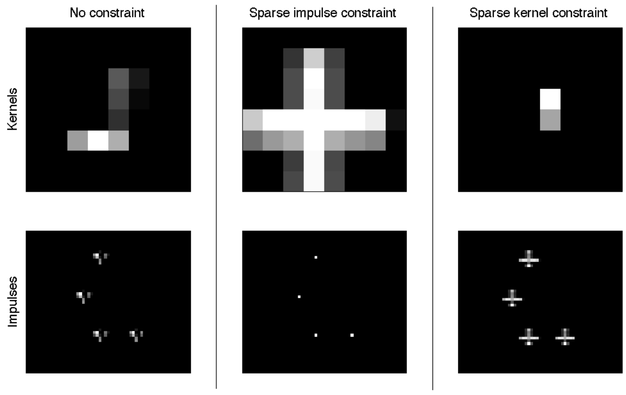

Fig. 5. Example of the effect of the entropic prior on a set of kernel and impulse distributions. If no constraint is imposed the information is evenly distributed among the two distributions (left column), if sparsity is imposed on the impulse distribution, most information lies in the kernel distribution (middle column), and vice verse if we request a sparse kernel distribution (right column).

figure 4. Other examples of such decompositions on more complex data are shown in [23].

The example above illustrates shift-invariance but it is conceivable that "components" that form the input might occur with transformations such as rotations and/or scaling in addition to translations (shifts). It is possible to extend this model to incorporate invariance to such transformations. The derivation follows naturally from the approach outlined above but we omit further discussion here due to space constraints.

C. Extensions in the form of Priors

One of the more apparent limitations of NMF is related to the quality of components that are extracted. Researchers have pointed out that NMF, as introduced by Lee and Seung, does not have an explicit way to control the "sparsity" of the desired components [26]. In fact, the inability to impose sparsity is just a specific example of a more general limitation. NMF does not provide a way to impose known or hypothesized structure about the data during estimation.

To elaborate, let us consider the example of sparsity. Several extensions have been proposed to NMF to incorporate sparsity (eg., [26], [27], [28]). The general idea in these methods is to impose a cost function during estimation that incorporates an additional constraint that quantifies the sparsity of the obtained factors. While sparsity is usually specified as the $L0$ norm of the derived factors [29], the actual constraints used consider an $L1$ norm, since the $L0$ norm is not amenable to optimization within a procedure that primarily attempts to minimize the $L2$ norm of the error between the original data and the approximation given by the estimated factors. In the probabilistic formulation the relationship of the sparsity constraint to the actual objective function optimized is more direct. We characterize sparsity through the entropy of the derived factors, as originally specified in [30]. A sparse code is defined as a set of basis vectors such that any given data point can be largely explained by only a few bases from the set, such that the required contribution of the rest of the bases to the data point is minimal; *i.e.* the entropy of the mixture weights by which the bases are combined to explain the data point is low. A sparse code can now be obtained by imposing the *entropic prior* over the mixture weights. For a given distribution $\boldsymbol{\theta}$, the entropic prior is defined as $P(\boldsymbol{\theta}) \propto e^{-\beta \mathcal{H}(\boldsymbol{\theta})}$ where $\mathcal{H}(\boldsymbol{\theta})$ is the entropy. Imposition of this prior (with a positive β) on the mixture weights just means that we obtain solutions where mixture weights with low entropy are more likely to occur - a low entropy ensures that few entries of the vector are significant. Sparsity has been imposed in latent variable models by utilizing the entropic prior and has been shown to provide a better characterization of the data [17], [18], [23], [31]. Detailed derivation and estimation algorithms can be found in [17], [18]. Notice that priors can be imposed on any set of parameters during estimation.

Information theoretically, entropy is a measure of information content. One can consider the entropic prior as providing an explicit way to control the amount of "information content" desired on the components. We illustrate this idea using a simple shift-invariance case. Consider an image which is composed out of scattered plus sign characters. Upon analysis of that image we would expect the kernel distribution to be a "+", and the impulse distribution to be a set of delta functions placing it appropriately in space. However using the entropic prior we can distribute the amount of information from the kernel distribution to the impulse distribution or vice-versa. We show the results from this analysis in figure 5 in terms of three cases - where no entropic prior is used (left panels), where it is used to make the impulse sparse (mid panels), and where it is used to make the kernel sparse (right panels). In the left panels, information about the data is distributed both in the kernel (top) and in the impulse distribution (bottom). In the other two cases, we were able to concentrate all the information either in the kernel or in the impulse distribution by making use of the entropic prior.

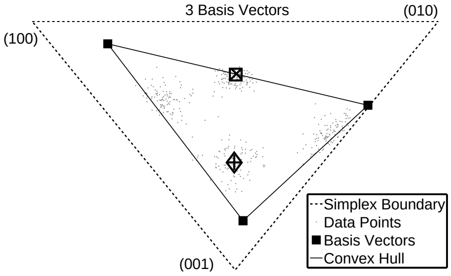

Fig. 6. Illustration of the latent variable model. Panel shows 3-dimensional data distributions as points within the *Standard 2-Simplex* given by $\{(001), (010), (100)\}$. The model approximates data distributions as points lying within the convex hull formed by the components (basis vectors). Also shown are two data points (marked by $+$ and $\times$) and their approximations by the model (respectively shown by $\diamondsuit$ and $\square$).

Other prior distributions that have been used in various contexts include the Dirichlet [8], [32] and log-normal distributions [33] among others. The ability to utilize prior distributions during estimation provides a way to incorporate information known about the problem. More importantly, the probabilistic framework provides proven methods of statistical inference techniques that one can employ for parameter estimation. We point out that these extensions can work with all the generalizations that were presented in the previous sections.

D. Geometrical Interpretation

We also want to briefly point out that probabilistic models can sometimes provide insights that are helpful for an intuitive understanding of the workings of the model.

Consider the asymmetric factorization case of the latent variable model as given by equation (7). Let us refer to the normalized columns of the data matrix $\mathbf{V}$ (obtained by scaling the entries of every column to sum to 1), $\bar{\mathbf{v}}_n$, as *data distributions*. It can be shown that learning the model is equivalent to estimating parameters such that the model $P(x_1|x_2)$ for any data distribution $\bar{\mathbf{v}}_{x_2}$ best approximates it. Notice that the data distributions $\bar{\mathbf{v}}_{x_2}$, model approximations $P(x_1|x_2)$, and components $P(x_1|z)$ are all M-dimensional vectors that sum to unity, and hence points in a $(M-1)$ simplex. The model expresses $P(x_1|x_2)$ as points within the convex hull formed by the components $P(x_1|z)$. Since it is constrained to lie within this convex hull, $P(x_1|x_2)$ can model $\bar{\mathbf{v}}_{x_2}$ accurately only if the latter also lies within the convex hull. Thus, the objective of the model is to estimate $P(x_1|z)$ as corners of a convex hull such that all the data distributions lie within. This is illustrated in Figure 6 for a toy dataset of 400 three-dimensional data distributions.

Not all probabilistic formulations provide such a clean geometric interpretation but in certain cases as outlined above, it can lead to interpretations that are intuitively helpful.

V. Discussion and Conclusions

In this paper we presented a family of latent variable models and shown their utility in the analysis of non-negative data. We show that the latent variable models decompositions are numerically identical to the NMF algorithm that optimizes a Kullback Leibler metric. Unlike previously reported results [34], the proof of equivalence requires no assumption about the distribution of the data, or indeed any assumption about the data besides non-negativity. The algorithms presented in this paper primarily compute a probabilistic factorization of non-negative data that optimizes the KL distance between the factored approximation and the actual data [2]. We argue that the use of this approach presents a much more straightforward way to make easily extensible models.

To demonstrate this we presented extensions that deal with tensorial data, shift-invariances and use priors on the estimation. The purpose of this paper is not to highlight the use of these approaches nor to present them thoroughly, but rather demonstrate a methodology which allows easier experimentation with non-negative data analysis and opens up possibilities for more stringent and probabilistic modeling than before. A rich variety of real-world applications and derivations of these and other models can be found in the references.

Acknowledgements

Madhusudana Shashanka acknowledges the support and helpful feedback received from Michael Giering at Mars, Inc.

References

[1] M. Plumbley and E. Oja, “A nonnegative pca algorithm for independent component analysis,” *IEEE Transactions on Neural Networks*, 2004.

[2] M. Plumbley, “Geometrical methods for non-negative ica: Manifolds, lie groups and toral subalgebras,” *Neurocomputing*, 2005.

[3] D. Lee and H. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, 1999.

[4] M. Heiler and C. Schnoerr, “Controlling sparseness in non-negative tensor factorization,” in *ECCV*, 2006.

[5] A. Cichocki, R. Zdunek, S. Choi, R. Plemmons, and S. Amari, “Non-negative tensor factorization using alpha and beta divergences,” in *ICASSP*, 2007.

[6] A. Shashua and T. Hazan, “Non-negative tensor factorization with applications to statistics and computer vision,” in *ICML*, 2005.

[7] T. Hofmann, “Unsupervised learning by probabilistic latent semantic analysis,” *Machine Learning*, vol. 42, pp. 177–196, 2001.

[8] D. Blei, A. Ng, and M. Jordan, “Latent dirichlet allocation,” *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[9] P. Lazarsfeld and N. Henry, *Latent Structure Analysis*. Boston: Houghton Mifflin, 1968.

[10] J. Rost and R. Langeheine, Eds., *Applications of Latent Trait and Latent Class Models in the Social Sciences*. New York: Waxmann, 1997.

[11] L. Goodman, “Exploratory latent structure analysis using both identifiable and unidentifiable models,” *Biometrika*, vol. 61, pp. 215–231, 1974.

[12] D. Lee and H. Seung, “Algorithms for non-negative matrix factorization,” in *NIPS*, 2001.

[13] B. Green Jr., “Latent structure analysis and its relation to factor analysis,” *Journal of the American Statistical Association*, vol. 47, pp. 71–76, 1952.

[14] P. Smaragdis and B. Raj, “Shift-invariant probabilistic latent component analysis,” *Journal of Machine Learning Research (under review)*, 2008.

[15] P. Smaragdis, B. Raj, and M. Shashanka, “Supervised and semi-supervised separation of sounds from single-channel mixtures,” in *Intl Conf on ICA and Signal Separation*, 2007.

[16] ——, “A probabilistic latent variable model for acoustic modeling,” in *NIPS Workshop on Advances in Modeling for Acoustic Processing*, 2006.

[17] M. Shashanka, B. Raj, and P. Smaragdis, “Sparse overcomplete latent variable decomposition of counts data,” in *NIPS*, 2007.

[2] It is not clear that the approach can be extended to similarly derive factorizations that optimize other Bregman divergences such as the $L2$ metric – this is a topic for further investigation

[18] M. Shashanka, “Latent variable framework for modeling and separating single-channel acoustic sources,” Ph.D. dissertation, Boston University, 2007.

[19] E. Gaussier and C. Goutte, “Relation between plsa and nmf and implications,” in *Proc. ACM SIGIR Conf. on Research and Dev. in Information Retrieval*, 2005, pp. 601–602.

[20] B. Raj and P. Smaragdis, “Latent variable decomposition of spectrograms for single channel speaker separation.” in *IEEE WASPAA*, 2005.

[21] M. Shashanka, B. Raj, and P. Smaragdis, “Probabilistic latent variable model for sparse decompositions of non-negative data,” *IEEE Trans on Pattern Analysis and Machine Intelligence (under review)*, 2008.

[22] M. Welling and M. Weber, “Positive tensor factorization,” *Pattern Recognition Letters*, 2001.

[23] P. Smaragdis, B. Raj, and M. Shashanka, “Sparse and shift-invariant feature extraction from non-negative data,” in *Intl Conf on Acoustics, Speech and Signal Processing*, 2008.

[24] P. Smaragdis, “Non-negative matrix factor deconvolution: Extraction of multiple sound sources from monophonic inputs,” in *Intl Conf on ICA and Blind Signal Separation*, 2004.

[25] ——, “Convolutive speech bases and their application to supervised speech separation,” *IEEE Trans on Audio, Speech and Language Processing*, 2007.

[26] P. Hoyer, “Non-negative matrix factorization with sparseness constraints,” *Journal of Machine Learning Research*, vol. 5, 2004.

[27] M. Morup and M. Schmidt, “Sparse non-negative matrix factor 2-d deconvolution,” Technical University of Denmark, Tech. Rep., 2006.

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

[29] D. Donoho, “For most large undetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 7, pp. 903–934, 2006.

[30] B. Olshausen and D. Field, “Emergence of simple-cell properties by learning a sparse code for natural images,” *Natrure*, vol. 381, 1996.

[31] M. Shashanka, B. Raj, and P. Smaragdis, “Sparse overcomplete decomposition for single channel speaker separation,” in *ICASSP*, 2007.

[32] B. Raj, M. Shashanka, and P. Smaragdis, “Latent dirichlet decomposition for single channel speaker separation,” in *ICASSP*, 2006.

[33] D. Blei and J. Lafferty, “Correlated topic models,” in *NIPS*, 2006.

[34] J. Canny, “Gap: A factor model for discrete data,” in *SIGIR*, 2004.