Probabilistic Factorization of Non-Negative Data with Entropic Co-occurrence Constraints

Paris Smaragdis[1], Madhusudana Shashanka[2], Bhiksha Raj[3], and Gautham J. Mysore[4⋆]

[1] Adobe Systems Inc.
[2] Mars Incorporated
[3] Mitsubishi Electric Research Laboratories
[4] Stanford University

Abstract. In this paper we present a probabilistic algorithm which factorizes non-negative data. We employ entropic priors to additionally satisfy that user specified pairs of factors in this model will have their cross entropy maximized or minimized. These priors allow us to construct factorization algorithms that result in maximally statistically different factors, something that generic non-negative factorization algorithms cannot not explicitly guarantee. We further show how this approach can be used to discover clusters of factors which allow a richer description of data while still effectively performing a low rank analysis.

1 Introduction

With the recent interest in non-negative factorization algorithms we have seen a rich variety of algorithms that can perform this task for a wide range of applications using various models. Empirically it has been observed that the non-negativity constraint in conjunction with the information bottleneck that such a low-rank factorization imposes, often results in data which is often interpreted as somewhat independent. Although this is approximately and qualitatively a correct observation, it is not something that is explicitly enforced in such algorithms and thus more a result of good fortune than planning. Nevertheless this property has proven to be a primary reason for the continued interest in such factorization algorithms. The task of finding independence in non-negative data has been explicitly tackled in the past using non-negative ICA and and PCA algorithms [1, 2] but such models have not been as easy to manipulate and extend as non-negative factorization models, which resulted in a diminished use of explicit independence optimization for non-negative data.

In this paper we present a non-negative factorization approach that explicitly manipulates the statistical relationships between the estimated factors. We recast the task of non-negative factorization as a probabilistic latent variable decomposition on count/histogram data. Using this abstraction we treat the input data as a multidimensional probability distribution and estimate an additive

⋆ Work performed while at Adobe Systems Inc

set of marginal distributions which would approximate it. This approach allows us to implicitly satisfy the non-negativity constraint (due to the fact that we estimate the factors as distributions), and at the same time allows for a convenient handle on statistical manipulations. In this paper we extend the original PLCA model [3], so that we can manipulate the cross entropy between the estimated marginals. This allows us to extract marginal distributions which are pairwise either similar or dissimilar. We also show how this approach can help in constructing more sophisticated analysis structures by enforcing the creation of related cliques of factors.

2 The PLCA model

Probabilistic Latent Component Analysis (PLCA) decomposes a multidimensional distribution as a mixture of latent components where each component is given by the product of one-dimensional marginal distributions. Although we will proceed by formulating the PLCA model using a two-dimensional input, this can be easily extended to inputs of arbitrary dimensions which can be seen as non-negative tensors. Given a two-dimensional input distribution $P(x_1, x_2)$, PLCA can be formulated as

$$P(x_1, x_2) = \sum_{z \in \boldsymbol{\mathcal{Z}}} P(z)P(x_1|z)P(x_2|z), \tag{1}$$

where z is a latent variable that indexes the latent components and takes values from the set $\boldsymbol{\mathcal{Z}} = \{z_1, z_2, \ldots, z_K\}$. Given a data matrix $\mathbf{V}$, parameters can be estimated by maximizing the log-likelihood given by

$$\mathcal{L} = \sum_{x_1, x_2} V_{x_1,x_2} \sum_{z} P(z|x_1, x_2) \log \Big[P(z)P(x_1|z)P(x_2|z) \Big]. \tag{2}$$

Iterative update equations obtained by using the EM algorithm are given by:

$$P(z|x_1, x_2) = \frac{P(z)P(x_1|z)P(x_2|z)}{\sum_{z \in \boldsymbol{\mathcal{Z}}} P(z)P(x_1|z)P(x_2|z)} \tag{3}$$

$$P(x_1|z) = \frac{\sum_{x_2} V_{x_1,x_2} P(z|x_1, x_2)}{\sum_{x_1,x_2} V_{x_1,x_2} P(z|x_1, x_2)}$$

$$P(x_2|z) = \frac{\sum_{x_1} V_{x_1,x_2} P(z|x_1, x_2)}{\sum_{x_1,x_2} V_{x_1,x_2} P(z|x_1, x_2)}$$

$$P(z) = \frac{\sum_{x_1,x_2} V_{x_1,x_2} P(z|x_1, x_2)}{\sum_{z,x_1,x_2} V_{x_1,x_2} P(z|x_1, x_2)}, \tag{4}$$

where equation (3) represents the Expectation step and equations (4) represents the Maximization step of the EM algorithm. As shown in [3], the above formulation can be expressed as a matrix factorization, where $P(x_1, x_2)$ represents a non-negative matrix and $P(x_1|z)$ and $P(x_2|z)$ are the factors along each of the input's dimensions.

3 Imposing Cross-Factor Constraints

In this section we describe how we can manipulate a statistical relationship between two arbitrary sets of marginal distributions in our model. For simplicity we will manipulate the relationship between two sets of marginals as observed along the dimension of x_1. Extending this to other or more dimensions is a trivial extension of the following formulation.

Let the two sets of latent variables be represented by $\boldsymbol{\mathcal{Z}}_1$ and $\boldsymbol{\mathcal{Z}}_2$ where $\boldsymbol{\mathcal{Z}}_1 \cup \boldsymbol{\mathcal{Z}}_2 \subseteq \boldsymbol{\mathcal{Z}}$. To impose our desired constraint, we need to make $P(x_1|z_1)$ and $P(x_1|z_2)$ similar or dissimilar from each other. We can achieve this by modifying the cost function of equation (2) by appending a metric that corresponds to the dissimilarity between the distributions. During estimation, maximizing or minimizing that cost function will result in biasing the estimation towards the desired outcome.

One measure we can use to describe the similarity between two distributions is the cross entropy. For two distributions $\mathbf{q}_{z_i} = P(x_1|z_i)$ and $\mathbf{p}_{z_k} = P(x_1|z_k)$, cross entropy is given by

$$H(\mathbf{q}_{z_i}, \mathbf{p}_{z_k}) = -\sum_{x_1} P(x_1|z_i) \log P(x_1|z_k). \tag{5}$$

Appending to the log-likelihood $\mathcal{L}$ cross-entropies $H(\mathbf{q}_{z_i}, \mathbf{p}_{z_k})$ and $H(\mathbf{p}_{z_k}, \mathbf{q}_{z_i})$ for all $z_i \in \boldsymbol{\mathcal{Z}}_1$ and $z_k \in \boldsymbol{\mathcal{Z}}_2$, we obtain the new cost function as[1]

$$\begin{aligned} Q &= \mathcal{L} + \alpha \sum_{i|z_i \in \boldsymbol{\mathcal{Z}}_1} \sum_{k|z_k \in \boldsymbol{\mathcal{Z}}_2} (H(\mathbf{q}_{z_i}, \mathbf{p}_{z_k}) + H(\mathbf{p}_{z_k}, \mathbf{q}_{z_i})) \\ &= \mathcal{L} - \alpha \sum_{i|z_i \in \boldsymbol{\mathcal{Z}}_1} \sum_{k|z_k \in \boldsymbol{\mathcal{Z}}_2} \sum_{t} P(x_1|z_k) \log P(x_1|z_i) \\ &\quad -\alpha \sum_{i|z_i \in \boldsymbol{\mathcal{Z}}_1} \sum_{k|z_k \in \boldsymbol{\mathcal{Z}}_2} \sum_{t} P(x_1|z_i) \log P(x_1|z_k) \end{aligned}$$

where α is a tunable parameter that controls the extent of regularization.

We use the EM algorithm again to estimate all the parameters. The E-step remains the same as given by the equation (3). Since the terms appended to $\mathcal{L}$ in the new cost function does not involve $P(f|z)$ or $P(z)$, the update equations for them remain the same as given by equations (4).

Consider the estimation of $\mathbf{q}_{z_i} = P(x_1|z_i)$ for a given value of i. Adding a Lagrange multiplier term and differentiating the new cost function with respect to $P(x_1|z_i)$ and setting it to 0, we obtain

$$\frac{\sum_{x_2} V_{x_1,x_2} P(z_i|x_1,x_2) - \alpha \sum_{z_k} P(x_1|z_k)}{P(x_1|z_i)} - \alpha \sum_{z_k} \log P(x_1|z_k) + \lambda = 0 \tag{6}$$

[1] This cost function is equivalent the log-posterior obtained in a MAP formulation where the exponential of the cross-entropy is used as a prior.

which implies that

$$\sum_{x_2} V_{x_1,x_2} P(z_i|x_1,x_2) - \alpha \sum_{z_k} P(x_1|z_k) = P(x_1|z_i)\Big(\alpha \sum_{z_k} \log P(x_1|z_k) - \lambda\Big),$$

where λ is the Lagrange multiplier. Treating the term $\log P(x_1|z_k)$ as a constant and utilizing the fact that $\sum_{x_1} P(x_1|z_i) = 1$, we can sum the above equation with respect to x_1 to obtain

$$\sum_{x_1}\sum_{x_2} V_{x_1,x_2} P(z_i|x_1,x_2) - \alpha \sum_{x_1}\sum_{z_k} P(x_1|z_k) = \alpha \sum_{z_k} \log P(x_1|z_k) - \lambda.$$

Utilizing this result in equation (6), we obtain the update equation as[2]

$$P(x_1|z_i) = \frac{\sum_{x_2} V_{x_1,x_2} P(z_i|x_1,x_2) - \alpha \sum_{z_k} P(x_1|z_k)}{\sum_{x_2}\sum_{x_1} V_{x_1,x_2} P(z_i|x_1,x_2) - \alpha \sum_{x_1}\sum_{z_k} P(x_1|z_k)} \tag{7}$$

Since $P(x_1|z_k)$ is treated as a constant during the estimation of $P(x_1|z_i)$ (and similarly $P(x_1|z_i)$ treated as a constant while estimating $P(x_1|z_k)$), the updates for $P(x_1|z_i)$ and $P(x_1|z_k)$ should be alternated between iterations. The above update equation works well when we attempt to perform minimization of the cross-entropy between marginals. It does however present a problem when we attempt to perform cross-entropy maximization while attempting to produce maximally different marginal distributions. To do so we would use a positive α which can potentially result in the outcome of equation 7 to be negative-valued. This is of course an inappropriate estimate for a distribution and would violate its implicit non-negative nature. The easiest way to deal with this problem is to discard any negative valued estimates and replace them with zeroes. In other experiments, more rigorously motivated approaches such as those employed in discriminative training methods [5], which prevent negative probability estimates by employing additive correction terms were observed to result in no appreciable difference in estimates.

Finally we note that the estimation equations as presented can be very effective in imposing the cross-entropy prior, sometimes counteracting the fit to the data. In practical situations we found it best to progressively reduce the weight of the prior across the EM iterations.

Examples of cross entropy manipulation To show the performance of the above estimation rules let us consider a simple non-negative factorization problem. As the input we use the magnitude spectrogram of a drums recording shown in figure 3.1. In this input we can clearly see the different types of drums in the

[2] When α is positive, the update equation for $P(x_1|z_i)$ can also be derived as follows. We construct a lower bound Q' for Q by removing all the $H(\mathbf{q}_{z_i}, \mathbf{p}_{z_k})$ terms from Q. Since $Q' \leq Q$, we can estimate parameters by maximizing Q' instead of maximizing Q. Adding a lagrangian to Q' and taking derivatives w.r.t. $P(x_i|z_i)$, it can be easily shown that the resulting update equation is given by equation (7).

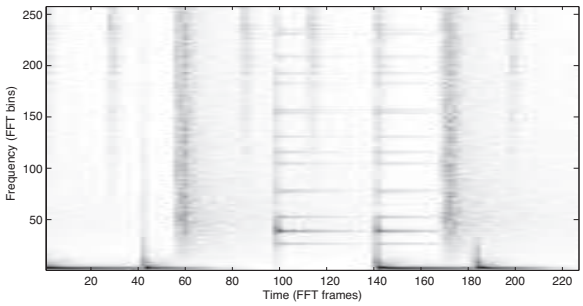

Fig. 3.1. The drums recording spectrogram we used as an input for the examples in this paper. In it one can clearly see each instrument and a factorization algorithm is expected to discover individual instruments by taking advantage of their distinct spectral and temporal positioning.

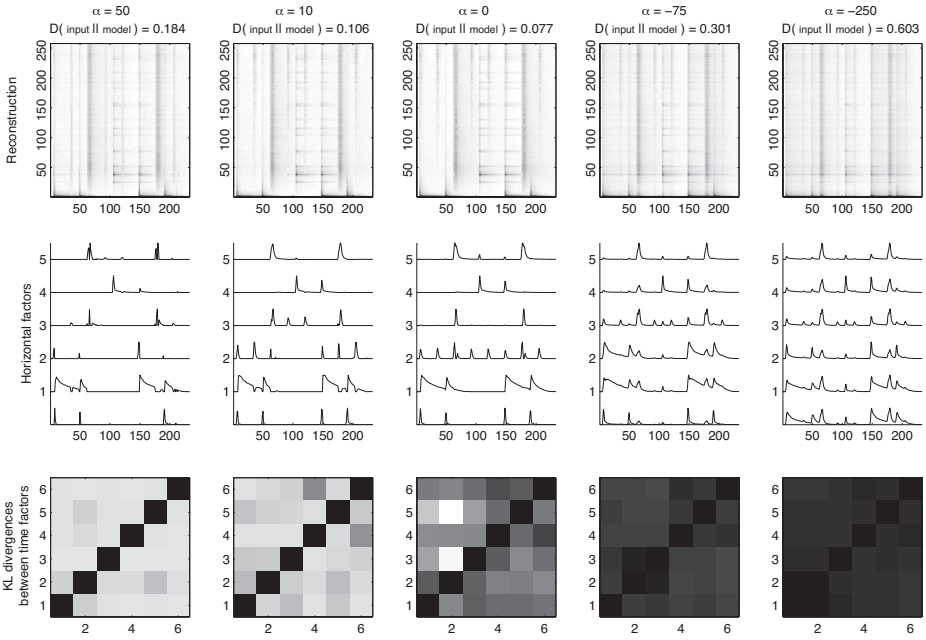

Fig. 3.2. The results of analyzing the same input with various prior weights. The examples from left to right column show the results with a maximal dissimilarity prior to a maximal similarity prior. The middle column is the case with no prior. The top row shows the reconstruction in each case, the middle row the extracted horizontal marginals and the bottom row their KL divergence.

mixture aided by their distinct spectral and temporal profiles. A factorization algorithm is expected to be able to distinguish between the various drum sounds for the same reasons. We performed the analysis using the same starting conditions but imposing a cross-entropy prior on the marginals corresponding to the time axis. The results of these analyses are shown in figure 3.2. The top row shows the reconstruction of the input, the middle row shows the extracted time marginals and the bottom row shows their mutual KL divergence (we chose to display the KL divergence since it is a more familiar indicator of relationship between distributions as opposed to the cross entropy). Each column of plots is a different analysis with the prior weight shown in the top. From left to right the prior goes from imposing maximal dissimilarity to maximal similarity. The middle column has no prior imposed. One can clearly see from the plots that for large positive values of the prior's weight we estimate a much more sparse set of marginals, and one where their respective KL divergence is maximal. As the prior weight is moved towards large negative values we gradually observe the discovery of less sparse codes, up to the extreme point where all extracted marginals are very similar.

4 Group-wise analysis

A powerful use of the above priors is that of performing a group-based analysis, similar to the concept of the multidimensional independent component analysis [4]. This means factorizing an input with a large number of components which are grouped in a smaller set of cliques of mutually related components. To perform this we need to partition the marginals of an analysis in groups and then use the prior we introduced to request minimal cross-entropy between the marginals in the same groups and maximal cross-entropy between the marginals from different groups. This will result in a collection of marginal groups in which elements of different groups are statistically different, whereas elements in the same group are similar.

To illustrate the practical implications of this approach consider the following experiment on the data of figure 3.1. We partitioned the twelve requested marginals in six groups of two. We performed the analysis with no priors, then with priors forcing the time marginals from separate groups to be different, then with priors forcing time marginals in the same group to be similar, and finally with both types of priors. All simulations were run with the same initial conditions. The results of these analyses are shown in figure 4.3. Like before each column shows different measures of the same analysis. The left most shows the case where no priors were used, the second one the case where the within group similarity was imposed, the third one where out of group dissimilarity was imposed and the rightmost one where both within group similarity and out of group dissimilarity were imposed. The top row shows the resulting reconstruction, the second row shows the discovered time marginals and the third row shows the KL divergence between the time marginals. Also shown in the titles of the top figures is the KL divergence between the input and the model reconstruction. Occasionally when we impose a prior in the model we observe a slight increase

which signifies that the model is not representing the input as accurately. Qualitatively this increase is usually fairly minor however and is expected since the model is not optimizing just the fit to the input data anymore. Observing the

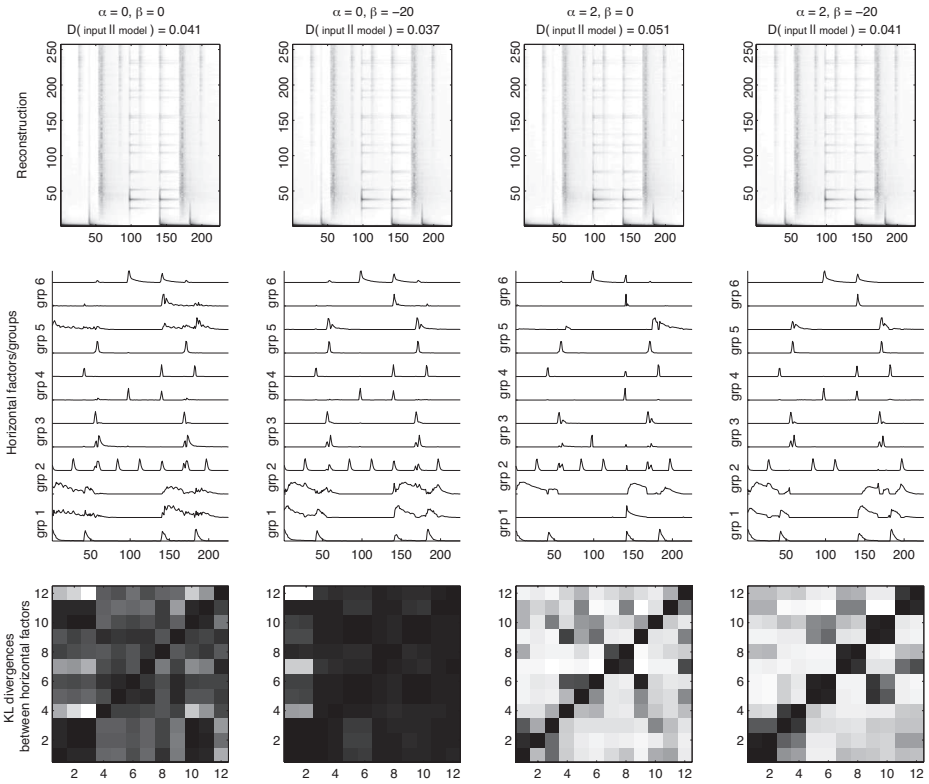

Fig. 4.3. Analysis results with various values of cross entropy priors. The variables α and β are the out-of-group dissimilarity and in-group similarity prior weights, and they were applied on the horizontal factors only. Each column of plots shows the results of a different analysis with its α and β values shown over the top figures of each column. The top figures show the resulting reconstructions and on their title we show the KL divergence between the input and the reconstruction. The middle figures show the resulting horizontal factors and the bottom figures show the KL divergence values between all factor pairs.

results we can clearly see that when we impose the within group similarity the extracted marginals belonging to the same group are more similar than otherwise. But by doing so we also implicitly encourage more similarity across all marginals since there is nothing to stop two different groups latching on the same instrument. In contrast when we use the out of group dissimilarity prior we see that we get very dissimilar marginals in the outputs, while some of them be-

longing in the same group happen to have some similarity. Imposing both of the previous priors at the same time results in a more desirable output where each group contains marginals which are dissimilar from marginals of other groups, yet similar to the marginals in the same group. Effectively we see the extracted marginals from the same groups latching on to different parts of the same drum sounds. Also shown at the top of each plot column is the KL divergence between the input and the model. We see that there is no significant deterioration in the fit when imposing these priors. More aggressive values of these priors will result in a worse fit, but an even stronger statistical separation (or not) between the extracted marginals.

In this particular case since the audio stream contains elements with time correlation we used that as the dimension in which the cross-entropy manipulation was performed. This would also be supplemented by using neighboring time samples in order to look for temporal causality as well. In other cases however we might prefer to impose priors on all dimensions as opposed to only one. The estimation process is flexible enough to deal with any number of dimensions and to impose priors that either minimize or maximize cross-entropy on any arbitrary dimension subset.

5 Conclusions

In this paper we presented a non-negative data factorization approach which allows us to discover factors which can be mutually similar or dissimilar. In order to do so we reformulated the nonnegative factorization process as a probabilistic factorization problem and introduced new priors that can minimize or maximize the cross-entropy between any of the discovered marginals. The cross-entropy was shown to be an appropriate measure of similarity which allows us to express arbitrary relationships between any of the estimated marginals. We've shown that using this approach we can perform analysis which is slightly more akin to ICA than NMF by extracting maximally different marginals, and also that we can extract groups of components which contain highly relevant marginals but bear little relation to other groups.

References

1. Plumbley, M.D. 2003. Algorithms for nonnegative independent component analysis, in IEEE Transactions on Neural Networks, Vol. 14: 3 May 2003.
2. Plumbley, M.D. and E. Oja. 2004. A "nonnegative PCA" algorithm for independent component analysis, in IEEE Transactions on Neural Networks, Vol. 15:1, Jan. 2004.
3. Shashanka, M., B. Raj, and P. Smaragdis. 2008. Probabilistic Latent Variable Models as Nonnegative Factorizations, in Computational Intelligence and Neuroscience, Volume 2008.
4. Cardoso, J-F. 1998. Multidimensional independent component analysis, in In Proceedings of the International Workshop on Higher-Order Statistics, 1998.
5. Schlüter, R., W. Macherey, B. Muëller, H. Ney, 2001. Comparison of discriminative training criteria and optimization methods for speech recognition, in Speech Communication, vol. 34 (2001) pp.287-310.