An Efficient Posterior Regularized Latent Variable Model for Interactive Sound Source Separation

Nicholas J. Bryan NJB@CCRMA.STANFORD.EDU
Center for Computer Research in Music and Acoustics, Stanford University

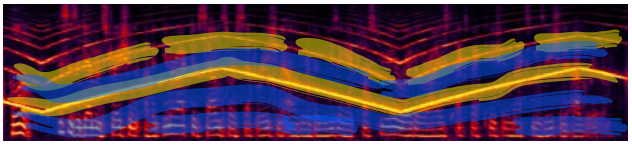
Gautham J. Mysore GMYSORE@ADOBE.COM
Adobe Research

Abstract

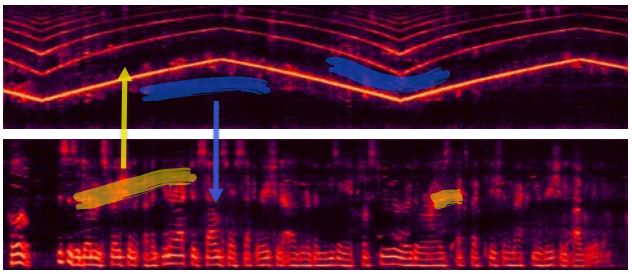
In applications such as audio denoising, music transcription, music remixing, and audio-based forensics, it is desirable to decompose a single-channel recording into its respective sources. One of the current most effective class of methods to do so is based on non-negative matrix factorization and related latent variable models. Such techniques, however, typically perform poorly when no isolated training data is given and do not allow user feedback to correct for poor results. To overcome these issues, we allow a user to interactively constrain a latent variable model by painting on a time-frequency display of sound to guide the learning process. The annotations are used within the framework of posterior regularization to impose linear grouping constraints that would otherwise be difficult to achieve via standard priors. For the constraints considered, an efficient expectation-maximization algorithm is derived with closed-form multiplicative updates, drawing connections to non-negative matrix factorization methods, and allowing for high-quality interactive-rate separation without explicit training data.

1. Introduction

Over the past several years, there has been a surge of research on single-channel sound source separation methods. Such methods focus on the task of separating a single monophonic recording of a mixture of sounds into its respective sources. The problem is motivated by many outstanding issues in signal processing and machine learning, such as speech denoising, speech enhancement, audio-based forensics, music transcription, and music remixing.



(a) A spectrogram of speech + siren with user annotations.



(b) Spectrograms of the initially separated speech and ambulance siren with further overlaid annotations.

Figure 1. (a) Given a mixture recording, a user separates distinct sounds by roughly painting on a time-frequency display. (b) Once initially separated, fine-tuning is performed by painting on the output results. Painting on one output track at a particular time-frequency point pushes the sound into the other track(s).

One of the most promising and effective class of approaches found for these purposes thus far is based on non-negative matrix factorization (NMF) (Lee & Seung, 2001; Smaragdis & Brown, 2003; Virtanen, 2007; Févotte et al., 2009) and its probabilistic latent variable model counterparts (Raj & Smaragdis, 2005; Smaragdis et al., 2006). These methods model spectrogram data or equivalently the magnitude of the short-time Fourier transform (STFT) of an audio recording as a linear combination of prototypical spectral com-

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

ponents over time. The prototypical spectral components and their gains are then used to separate out each source within the mixture.

In many cases, these methods can achieve good separation results using supervised or semi-supervised techniques, where isolated training data is used to learn individual models of distinct sound sources, and then separate an unknown mixture of similar sounding sources (Smaragdis et al., 2007). When no training data is available, however, the methods are not useable without further assumptions.

Initial work to overcome this issue has been proposed which allows a user to annotate a time-frequency display of sound to inform the separation process without training. In Durrieu et al. (2012), a user is asked to annotate the fundamental frequency on a pitch-based display to inform a non-negative source-filter model to remove vocals from background music. In Lefévre et al. (2012), a user is asked to annotate binary time-frequency patches to perform semi-supervised separation with the intention of using the annotations to train an automatic, user-free system. While promising, these methods motivate further work for more general, flexible, and powerful solutions. In particular, the first method is limited to separating a pitched source from background music and the second method only allows for binary time-frequency annotations, disallowing a user to express a confidence level in the annotations.

To overcome these issues, we propose a new source separation method to separate arbitrary sounds without explicit isolated training data. The method allows a user to interactively constrain a probabilistic latent variable model used for separation by roughly painting on a spectrogram display of sound as shown in Fig. 1. Once an initial separation is performed, further annotations are used to refine the outputs and iteratively improve results, akin to the interactive clustering work of Cohn et al. (2003). To incorporate the constraints, we use the framework of posterior regularization (PR) and derive in an efficient expectation-maximization (EM) algorithm with closed-form multiplicative updates that allows for interactive-rate separation. For evaluation, a user-interface was developed and tested on several mixture sounds, showing the proposed method can achieve state-of-the-art results without explicit training data.

2. Proposed Method

To perform separation, we build off of the symmetric probabilistic latent component analysis model proposed by Smaragdis et al. (2006; 2007) as discussed in Section 3. Instead of performing supervised or semi-supervised separation requiring the use of training data such as proposed by Smaragdis et al. (2007), we allow a user to weakly guide the separation process by interactively providing intuitive annotations that, in turn, control regularization parameters in our model. This technique allows us to perform separation in the scenario when no training data is available.

More specifically, we first allow a user to annotate time-frequency features within a mixture recording that appear to correspond to one source or another as shown in Fig. 1a, using color to denote source and opacity as a measure of confidence. We then perform an initial separation given the annotations and allow the user to listen to the separated output. If the results are unsatisfactory, the user can then annotate errors in the output estimates as shown in Fig. 1b, and iteratively re-run the process—interactively updating the separation estimates until a desired result is achieved.

To algorithmically achieve the proposed interaction, a new method of injecting constraints into our model as a function of time, frequency, and sound source is outlined in Section 4. Moreover, the method must allow for interactive-rate (on the order of seconds) separation, making the issue of computational cost central to our goal. As a result, the proposed approach is carefully designed with these requirements in mind. The complete separation process is then discussed in Section 5, with evaluation and conclusions in Section 6 and Section 7 respectively.

3. Probabilistic Model

Probabilistic latent component analysis (PLCA) is a straightforward extension of probabilistic latent semantic indexing (PLSI) or equivalently probabilistic latent semantic analysis (PLSA) (Hofmann, 1999) for arbitrary dimensions. The general PLCA model is defined as a factorized probabilistic latent variable model of the form

$$P(\mathbf{x}) = \sum_{z} P(z) \prod_{j=1}^{N} P(x_j|z) \tag{1}$$

where $P(\mathbf{x})$ is an N-dimensional distribution of a random variable $\mathbf{x} = x_1, x_2, ..., x_N$, $P(z)$ is the distribution of the latent variable z, $P(x_j|z)$ are one-dimensional distributions, and the parameters of the distributions $\mathbf{\Theta}$ are implicit in the notation.

When employed for source separation, typically a two-

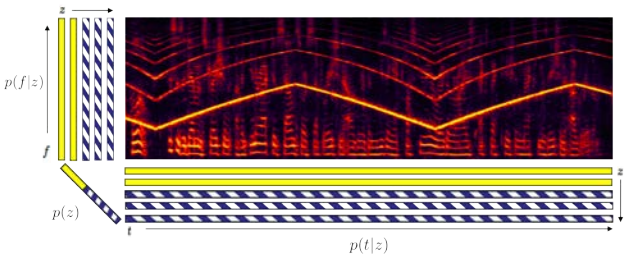


Figure 2. A probabilistic latent component analysis factorization of an audio spectrogram. Solid yellow elements of the distributions explain source A (e.g. siren), while blue striped elements explain source B (e.g. speech).

dimensional variant of the PLCA model

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

is used to approximate a normalized audio spectrogram $\mathbf{X}$, where the two-dimensions correspond to time and frequency ($f \equiv x_1$ and $t \equiv x_2$). The random variables f, t, and z are discrete and can take on N_f, N_t, and N_z possible values respectively. $P(f|z)$ is a multinomial distribution representing frequency basis vectors or dictionary elements for each source, and $P(t|z)$ and $P(z)$ are multinomial distributions, which together represent the weighting or activations of each frequency basis vector. N_z is typically chosen by a user and N_f and N_t are a function of the overall recording length and STFT parameters (transform length, zero-padding size, and hop size).

To model multiple sources N_s within a mixture, non-overlapping values of the latent variable are associated or grouped with each source and estimated using an expectation-maximization algorithm. Fig. 2 shows an example where two values of z are ideally associated with one source and the remaining three values to another, segmenting each distribution into two non-overlapping groups ($N_s = 2$ and $N_z = 2 + 3$). Unfortunately, such ideal segmentation rarely occurs, requiring supervised or semi-supervised methods (and isolated training data) to estimate $P(f|z)$ a priori for each source, motivating the proposed approach.

3.1. Parameter Estimation

Given our model and observed data $\mathbf{X}$, we can use an expectation-maximization (EM) algorithm to find a maximum likelihood solution to our model parameters $\mathbf{\Theta}$. We follow the standard approach of lower bounding the log-likelihood via

$$\ln P(\mathbf{X}\,|\mathbf{\Theta}) = \mathcal{F}(Q,\mathbf{\Theta}) + \mathrm{KL}(Q||P) \tag{3}$$

$$\mathcal{F}(Q,\mathbf{\Theta}) = \sum_{\mathbf{Z}} Q(\mathbf{Z}) \ln\left\{\frac{P(\mathbf{X},\mathbf{Z}\,|\mathbf{\Theta})}{Q(\mathbf{Z})}\right\} \tag{4}$$

$$\begin{aligned}\mathrm{KL}(Q||P) &= \mathrm{KL}(Q(\mathbf{Z}) \,||\, P(\mathbf{Z}\,|\,\mathbf{X},\mathbf{\Theta})) \\ &= -\sum_{\mathbf{Z}} Q(\mathbf{Z}) \ln\left\{\frac{P(\mathbf{Z}\,|\,\mathbf{X},\mathbf{\Theta})}{Q(\mathbf{Z})}\right\}\end{aligned} \tag{5}$$

for any discrete distribution $Q(\mathbf{Z})$, denoted by Q for compactness, where $\mathrm{KL}(Q||P)$ is the Kullback-Leibler divergence and $\mathcal{F}(Q,\mathbf{\Theta})$ is the lower bound as a result of $\mathrm{KL}(Q||P)$ being non-negative (Bishop, 2006).

With an initial guess of our model parameters, we then solve a two-stage coordinate ascent optimization. We first maximize the lower bound $\mathcal{F}(Q,\mathbf{\Theta})$ or equivalently minimize $\mathrm{KL}(Q||P)$ with respect to Q

$$\begin{aligned}Q^{n+1} &= \arg\max_Q \mathcal{F}(Q,\mathbf{\Theta}^n) \\ &= \arg\min_Q \mathrm{KL}(Q||P)\end{aligned} \tag{6}$$

and then maximize the lower bound with respect to $\mathbf{\Theta}$

$$\mathbf{\Theta}^{n+1} = \arg\max_{\mathbf{\Theta}} \mathcal{F}(Q^{n+1},\mathbf{\Theta}) \tag{7}$$

and repeat the process until convergence (the superscript ^{n} denotes the iteration). As known in the literature, such process guarantees parameter estimates $\mathbf{\Theta}$ to monotonically increase the lower bound $\mathcal{F}(Q,\mathbf{\Theta})$, and consequently the likelihood until convergence to a local stationary point. Also note that, in many cases, the expectation step only involves computing the posterior distribution $P(\mathbf{Z}\,|\,\mathbf{X},\mathbf{\Theta})$ because $Q(\mathbf{Z})$ is optimal when equal to the posterior, making it common to implicitly define $Q(\mathbf{Z})$. When we discuss the idea of posterior regularization below, however, an explicit representation of $Q(\mathbf{Z})$ is needed.

3.2. PLCA Algorithm

When we apply the above procedure to solve for the maximum likelihood parameters of our sound model, we get an iterative EM algorithm with closed-form updates at each iteration. The algorithm is outlined in Algorithm 1, where the subscript (f,t) is used to index $\mathbf{X}$ as a function of time and frequency. Given proper initialization and normalization, these update equations can be further rearranged using matrix notation (Smaragdis & Raj, 2007) are numerically identical to the multiplicative update equations for NMF with a KL divergence cost function as derived by Lee and Seung (2001).

Algorithm 2 shows the multiplicative update rules where $\mathbf{W}$ is a matrix of probability values such that

Algorithm 1 PLCA in Basic Form

Procedure PLCA-BASIC (
$\mathbf{X} \in \mathbf{R}_{+}^{N_f \times N_t}$, // observed normalized data
N_z // number of basic vectors
)
initialize: feasible $P(z)$, $P(f|z)$, and $P(t|z)$
repeat
expectation step
for all z, f, t **do**

$$Q(z|f,t) \leftarrow \frac{P(z)P(f|z)P(t|z)}{\sum_{z'} P(z')P(f|z')P(t|z')} \quad (8)$$

end for
maximization step
for all z, f, t **do**

$$P(f|z) \leftarrow \frac{\sum_t \mathbf{X}_{(f,t)} Q(z|f,t)}{\sum_{f'} \sum_{t'} \mathbf{X}_{(f',t')} Q(z|f',t')} \quad (9)$$

$$P(t|z) \leftarrow \frac{\sum_f \mathbf{X}_{(f,t)} Q(z|f,t)}{\sum_{f'} \sum_{t'} \mathbf{X}_{(f',t')} Q(z|f',t')} \quad (10)$$

$$P(z) \leftarrow \frac{\sum_f \sum_t \mathbf{X}_{(f,t)} Q(z|f,t)}{\sum_{z'} \sum_{f'} \sum_{t'} \mathbf{X}_{(f',t')} Q(z'|f',t')} \quad (11)$$

end for
until convergence
return: $P(f|z)$, $P(t|z)$, $P(z)$, and $Q(z|f,t)$

$P(f|z)$ is the f^{th} row and z^{th} column, $\mathbf{H}$ is a matrix of probability values such that $P(t|z)P(z)$ is the z^{th} row and t^{th} column, $\mathbf{1}$ is an appropriately sized matrix of ones, $\odot$ is element-wise multiplication, and the division is element-wise.

4. Posterior Regularization

Incorporating the user-annotations into our latent variable model can be done in several ways. As mentioned above, we need a method to incorporate grouping constraints as a function of source, time, and frequency. Given our factorized model, this is not easily accomplished using standard priors, motivating the use of posterior regularization, which is well suited for our task.

Posterior regularization for EM algorithms was first introduced by Graça, Ganchev, and Taskar (2007; 2009; 2009) as a way of injecting rich, typically data-dependent, constraints on the posterior distributions of latent variable models. The method has found success in many natural language processing tasks such as statistical word alignment, part-of-speech tagging, and similar tasks.

Algorithm 2 PLCA in Multiplicative Form

Procedure PLCA-MF (
$\mathbf{X} \in \mathbf{R}_{+}^{N_f \times N_t}$, // observed normalized data
N_z // number of basic vectors
)
initialize: feasible $\mathbf{W} \in \mathbf{R}_{+}^{N_f \times N_z}$ and $\mathbf{H} \in \mathbf{R}_{+}^{N_z \times N_t}$
repeat

$$\mathbf{Z} \leftarrow \frac{\mathbf{X}}{\mathbf{W}\,\mathbf{H}} \quad (12)$$

$$\mathbf{W} \leftarrow \mathbf{W} \odot \frac{\mathbf{Z}\,\mathbf{H}^{\mathrm{T}}}{\mathbf{1}\,\mathbf{H}^{\mathrm{T}}} \quad (13)$$

$$\mathbf{H} \leftarrow \mathbf{H} \odot (\mathbf{W}^{\mathrm{T}}\,\mathbf{Z}) \quad (14)$$

until convergence
return: $\mathbf{W}$ and $\mathbf{H}$

The basic idea is to constrain the distribution Q in some way when computing the expectation step of an EM algorithm. This can be seen by modifying the expectation step discussed in Section 3.1, resulting in

$$Q^{n+1} = \arg\min_{Q} \mathrm{KL}(Q||P) + \Omega(Q) \quad (15)$$

where $\Omega(Q)$ constrains the possible space of Q. To denote the use of constraints in this context, the term "weakly-supervised" was introduced by Graça (2009) and is similarly adopted here.

This method of regularization is in contrast to prior-based regularization, where the modified maximization step is

$$\mathbf{\Theta}^{n+1} = \arg\max_{\mathbf{\Theta}} \mathcal{F}(Q^{n+1}, \mathbf{\Theta}) + \Omega(\mathbf{\Theta}), \quad (16)$$

where $\Omega(\mathbf{\Theta})$ constrains the model parameters $\mathbf{\Theta}$. Now, given the general framework, we can introduce the specific form of the regularization used for our purpose.

4.1. Linear Grouping Expectation Constraints

To efficiently incorporate the user-annotated constraints into our latent variable model, we need to define a meaningful penalty $\Omega(Q)$. This is done by applying non-overlapping linear grouping constraints on the latent variable z, encouraging distinct groupings of the model factors to explain distinct sound sources. The strength of the constraints are then interactively tuned by a user as a function of the observed variables in our model f and t. As a result, we no longer can assign Q to simply be the posterior, and need to solve a separate constrained optimization problem.

To do so, we rewrite all values of Q and $P(z|f,t)$ for a given value of f and t in vector notation as $\mathbf{q}$ and $\mathbf{p}$,

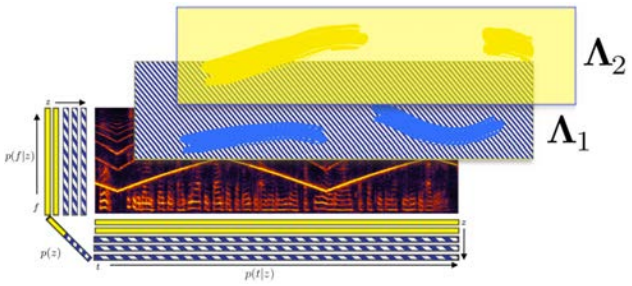


Figure 3. Probabilistic latent component analysis with user-annotated posterior regularization constraints.

and solve

$$\begin{aligned} \arg\min_{\mathbf{q}} \quad & -\mathbf{q}^{\mathrm{T}} \ln \mathbf{p} + \mathbf{q}^{\mathrm{T}} \ln \mathbf{q} + \mathbf{q}^{\mathrm{T}} \boldsymbol{\lambda} \\ \text{subject to} \quad & \mathbf{q}^{\mathrm{T}} \mathbf{1} = 1,\ \mathbf{q} \succeq 0 \end{aligned} \tag{17}$$

independently for each time-frequency ($N_f \times N_t$) value in our model at each expectation step. We then define $\boldsymbol{\lambda} \in \mathbf{R}^{N_z}$ as the vector of user-defined penalty weights, $^{\mathrm{T}}$ is a matrix transpose, $\succeq$ is element-wise greater than or equal to, and $\mathbf{1}$ is a column vector of ones.

To impose the penalties as a function of source, we partition the values of z to correspond to different sources or groups as described above and then set the corresponding penalty coefficients in $\boldsymbol{\lambda}$ to be identical within each group (e.g. $\boldsymbol{\lambda} = [\alpha, \alpha, \beta, \beta, \beta]$ for some $\alpha, \beta \in \mathbf{R}$). The entire set of real-valued grouping penalties are then defined as $\boldsymbol{\Lambda} \in \mathbf{R}^{N_f \times N_t \times N_z}$ indexed by frequency, time, and latent component or, alternatively, $\boldsymbol{\Lambda}_s \in \mathbf{R}^{N_f \times N_t}$, $\forall\ s \in \{1, .., N_s\}$, indexed by frequency, time, and source (group of latent components). Positive-valued penalties are used to decrease the probability of a given source, while negative-valued coefficients are used to increase the probability of a given source. Fig. 3 illustrates an example set of penalties $(\boldsymbol{\Lambda}_1, \boldsymbol{\Lambda}_2)$ as image overlays for two sources.

To solve the above optimization problem, we form the Lagrangian

$$\mathcal{L}(\mathbf{q}, \gamma) = -\mathbf{q}^{\mathrm{T}} \ln \mathbf{p} + \mathbf{q}^{\mathrm{T}} \ln \mathbf{q} + \mathbf{q}^{\mathrm{T}} \boldsymbol{\lambda} + \gamma(1 - \mathbf{q}^{\mathrm{T}} \mathbf{1})$$

with γ being a Lagrange multiplier, take the gradient with respect to $\mathbf{q}$ and γ

$$\begin{aligned} \nabla_{\mathbf{q}} \mathcal{L}(\mathbf{q}, \gamma) &= -\ln \mathbf{p} + \mathbf{1} + \ln \mathbf{q} + \boldsymbol{\lambda} - \gamma \mathbf{1} = 0 \quad (18) \\ \nabla_{\alpha} \mathcal{L}(\mathbf{q}, \gamma) &= (1 - \mathbf{q}^{\mathrm{T}} \mathbf{1}) = 0 \quad (19) \end{aligned}$$

set equations (18) and (19) equal to zero, and solve for $\mathbf{q}$, resulting in

$$\mathbf{q} = \frac{\mathbf{p} \odot \exp\{-\boldsymbol{\lambda}\}}{\mathbf{p}^{\mathrm{T}} \exp\{-\boldsymbol{\lambda}\}} \tag{20}$$

where $\exp\{\}$ is an element-wise exponential function. Notice the result is computed in closed-form and does not require any iterative optimization scheme as may be required in the general posterior regularization framework (Graça et al., 2007), limiting the computational cost when incorporating the constraints as our design objective requires.

4.2. Posterior Regularized PLCA

Knowing the posterior-regularized expectation step optimization, we can derive a complete EM algorithm for a posterior-regularized two-dimensional PLCA model (PR-PLCA). The modification becomes only a small change to the original PLCA algorithm, which replaces equation (8) with

$$Q(z|f,t) \leftarrow \frac{P(z)P(f|z)P(t|z)\tilde{\boldsymbol{\Lambda}}_{(f,t,z)}}{\sum_{z'} P(z')P(f|z')P(t|z')\tilde{\boldsymbol{\Lambda}}_{(f,t,z')}} \tag{21}$$

where $\tilde{\boldsymbol{\Lambda}} = \exp\{-\boldsymbol{\Lambda}\}$. The entire algorithm is outlined in Algorithm 3. Notice, we continue to maintain closed-form E and M steps, allowing us to optimize further and draw connections to multiplicative non-negative matrix factorization algorithms.

4.3. Multiplicative Update Equations

To compare the proposed method to the multiplicative form of the PLCA algorithm outlined in Algorithm 2, we can rearrange the expressions in Algorithm 3 and convert to a multiplicative form following similar methodology to Smaragdis and Raj (2007). Rearranging the expectation and maximization steps, in conjunction with Bayes' rule, and $Z(f,t) = \sum_z P(z)P(f|z)P(t|z)\tilde{\boldsymbol{\Lambda}}_{(f,t,z)}$, we get

$$\begin{aligned} Q(z|f,t) &= \frac{P(f|z)P(t,z)\tilde{\boldsymbol{\Lambda}}_{(f,t,z)}}{Z(f,t)} \quad (22) \\ P(t,z) &= \sum_f \mathbf{X}_{(f,t)}\, Q(z|f,t) \quad (23) \\ P(f|z) &= \frac{\sum_t \mathbf{X}_{(f,t)}\, Q(z|f,t)}{\sum_t P(t,z)} \quad (24) \\ P(z) &= \sum_t P(t,z) \quad (25) \end{aligned}$$

Rearranging further, we get

$$P(f|z) = \frac{P(f|z) \sum_t \frac{\mathbf{X}_{(f,t)}\, \tilde{\boldsymbol{\Lambda}}_{(f,t,z)}}{Z(f,t)} P(t,z)}{\sum_t P(t,z)} \tag{26}$$

$$P(t,z) = P(t,z) \sum_f P(f|z) \frac{\mathbf{X}_{(f,t)}\, \tilde{\boldsymbol{\Lambda}}_{(f,t,z)}}{Z(f,t)} \tag{27}$$

Algorithm 3 PR-PLCA with Linear Grouping Expectation Constraints in Basic Form

Procedure PR-PLCA-BASIC (
$\mathbf{X} \in \mathbf{R}_{+}^{N_f \times N_t}$, // observed normalized data
N_z, // number of basic vectors
N_s // number of sources
$\mathbf{\Lambda} \in \mathbf{R}^{N_f \times N_t \times N_z}$ // penalties
)
initialize: feasible $P(z)$, $P(f|z)$, and $P(t|z)$
precompute: $\tilde{\mathbf{\Lambda}} \leftarrow \exp\{-\mathbf{\Lambda}\}$
repeat
expectation step
for all z, f, t **do**

$$Q(z|f,t) \leftarrow \frac{P(z)P(f|z)P(t|z)\tilde{\mathbf{\Lambda}}_{(f,t,z)}}{\sum_{z'} P(z')P(f|z')P(t|z')\tilde{\mathbf{\Lambda}}_{(f,t,z')}} \quad (28)$$

end for
maximization step
for all z, f, t **do**

$$P(f|z) \leftarrow \frac{\sum_t \mathbf{X}_{(f,t)} Q(z|f,t)}{\sum_{f'} \sum_{t'} \mathbf{X}_{(f',t')} Q(z|f',t')} \quad (29)$$

$$P(t|z) \leftarrow \frac{\sum_f \mathbf{X}_{(f,t)} Q(z|f,t)}{\sum_{f'} \sum_{t'} \mathbf{X}_{(f',t')} Q(z|f',t')} \quad (30)$$

$$P(z) \leftarrow \frac{\sum_f \sum_t \mathbf{X}_{(f,t)} Q(z|f,t)}{\sum_{z'} \sum_{f'} \sum_{t'} \mathbf{X}_{(f',t')} Q(z'|f',t')} \quad (31)$$

end for
until convergence
return: $P(f|z)$, $P(t|z)$, $P(z)$, and $Q(z|f,t)$

Algorithm 4 PR-PLCA with Linear Grouping Expectation Constraints in Multiplicative Form

Procedure PR-PLCA-MF (
$\mathbf{X} \in \mathbf{R}_{+}^{N_f \times N_t}$, // observed normalized data
N_z, // number of basic vectors
N_s // number of sources
$\mathbf{\Lambda}_s \in \mathbf{R}^{N_f \times N_t}$, $\forall\, s \in \{1, .., N_s\}$ // penalties
)
initialize: feasible $\mathbf{W} \in \mathbf{R}_{+}^{N_f \times N_z}$ and $\mathbf{H} \in \mathbf{R}_{+}^{N_z \times N_t}$
precompute:
for all s **do**

$$\tilde{\mathbf{\Lambda}}_s \leftarrow \exp\{-\mathbf{\Lambda}_s\} \quad (32)$$

$$\mathbf{X}_s \leftarrow \mathbf{X} \odot \tilde{\mathbf{\Lambda}}_s \quad (33)$$

end for
repeat

$$\mathbf{\Gamma} \leftarrow \sum_s (\mathbf{W}_{(s)}\, \mathbf{H}_{(s)}) \odot \tilde{\mathbf{\Lambda}}_s \quad (34)$$

for all s **do**

$$\mathbf{Z}_s \leftarrow \frac{\mathbf{X}_s}{\mathbf{\Gamma}} \quad (35)$$

$$\mathbf{W}_{(s)} \leftarrow \mathbf{W}_{(s)} \odot \frac{\mathbf{Z}_s\, \mathbf{H}_{(s)}{}^{\mathrm{T}}}{\mathbf{1}\, \mathbf{H}_{(s)}{}^{\mathrm{T}}} \quad (36)$$

$$\mathbf{H}_{(s)} \leftarrow \mathbf{H}_{(s)} \odot (\mathbf{W}_{(s)}{}^{\mathrm{T}}\, \mathbf{Z}_s) \quad (37)$$

end for
until convergence
return: $\mathbf{W}$ and $\mathbf{H}$

which fully specifies the iterative updates. By putting equations (26) and (27) in matrix notation, we specify the multiplicative form of the proposed method in Algorithm 4. The subscript notation (s) with parenthesis is used as an index operator that picks off the appropriate column or rows of a matrix assigned to a given source, and the subscript s without parenthesis as an enumeration of similar variables.

4.4. Computational Cost

Neglecting the pre-computation step in Algorithm 4, we consider the increase in computational cost at each EM iteration of the proposed method over the standard PLCA update equations in Algorithm 2. We notice that only equations (34) and (35) add computation compared to their counterpart of equation (12) in Algorithm 2 as a result of careful indexing of equations (36) and (37). Additionally, equation (12) of Algorithm 2 consists of an $O(N_f N_t N_z)$ matrix multiplication and an $O(N_f N_t)$ element-wise matrix division.

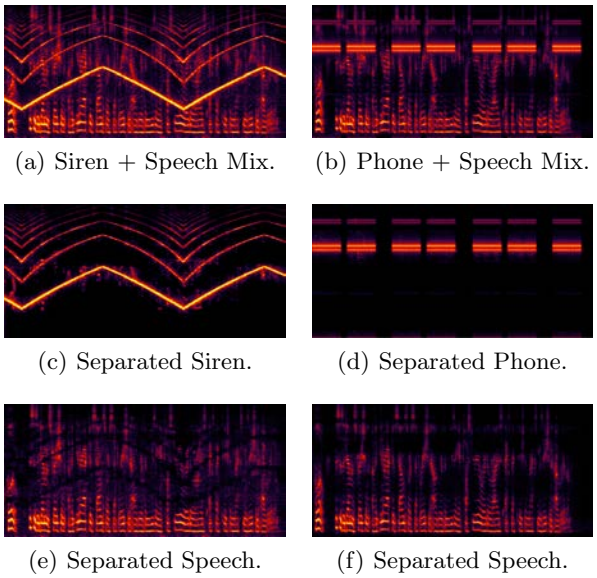
In contrast, equations (34) and (35) of Algorithm 4 consist of an $O(N_f N_t N_z)$ matrix multiplication, and an $O(N_s N_f N_t)$ element-wise matrix multiplication, division, and addition. In total, the difference is only an $O(N_f N_t N_s)$ element-wise matrix multiplication, division, and addition per EM iteration. As a result, the entire added cost per EM iteration for small N_s (typically two) is low and found to be acceptable in practice.

5. Compete Separation Process

To perform the complete separation process, we need to run Algorithm 4 in conjunction with pre- and post-computation. This involves first computing the short-time Fourier transform of the mixture recording, eliciting user-annotated penalties, running Algorithm 4, and then reconstructing the distinct sound sources from the output. To reconstruct the distinct sources from the output, we take the output posterior distribution and compute the overall probability of each source $p(s|f,t)$. This is done by summing

Algorithm 5 Complete PR-PLCA Source Separation

Procedure PR-PLCA-SEPARATION (
$\mathbf{x} \in \mathbf{R}^{N_\tau}$, // time-domain mixture signal
N_z, // number of basic vectors
N_s, // number of sources
P // STFT parameters
)
precompute:
$(\mathbf{X}, \angle \mathbf{X}) \leftarrow \text{STFT}(\mathbf{x}, P)$
repeat
input: user-annotated penalties
$\mathbf{\Lambda}_s \in \mathbf{R}^{N_f \times N_t}, \ \forall \ s \in \{1, .., N_s\}$
$(\mathbf{W}, \mathbf{H}) \leftarrow \text{PR-PLCA-MF}(\mathbf{X}, \mathbf{\Lambda}_s \ \forall \ s, N_z, N_s)$
for all s **do**
$\mathbf{M}_s \leftarrow \mathbf{W}_{(s)} \mathbf{H}_{(s)} \,/\, \mathbf{W}\,\mathbf{H}$ // compute filter
$\hat{\mathbf{X}}_s \leftarrow \mathbf{M}_s \odot \mathbf{X}$ // filter mixture
$\mathbf{x}_s \leftarrow \text{ISTFT}(\hat{\mathbf{X}}_s, \angle \mathbf{X}, \text{P})$
end for
until satisfied
return: time-domain signals $\mathbf{x}_s, \ \forall s \in \{1, ..., N_s\}$



(a) Siren + Speech Mix. (b) Phone + Speech Mix.
(c) Separated Siren. (d) Separated Phone.
(e) Separated Speech. (f) Separated Speech.

Figure 4. Two mixture spectrograms and the resulting separated sources using the proposed method for five minutes.

over the values of z that correspond to the source $P(s|f,t) = \sum_{z \in s} P(z|f,t)$ or equivalently by computing $\mathbf{W}_{(s)} \mathbf{H}_{(s)} \,/\, \mathbf{W}\,\mathbf{H}$. The probability of each source is then used to filter the mixture recording by element-wise multiplication with the input mixture spectrogram $\mathbf{X}$ according to standard practice (Benaroya et al., 2003). The result is then converted to a time-domain audio signal via an inverse STFT using the input mixture phase $\angle \mathbf{X}$.

The complete method is outlined in Algorithm 5, where we additionally define the forward short-time Fourier transforms $(\mathbf{X}, \angle \mathbf{X}) \leftarrow \text{STFT}(\mathbf{x}, P)$ as an algorithm that inputs a time-domain mixture signal $\mathbf{x}$ and STFT parameters P and returns the magnitude $\mathbf{X}$ matrix and phase matrix $\angle \mathbf{X}$. The inverse short-time Fourier transform $\mathbf{x} \leftarrow \text{ISTFT}(\mathbf{X}, \angle \mathbf{X}, P)$ then inputs a magnitude matrix, phase matrix, and parameters P and returns a time-domain signal $\mathbf{x}$. For a reference on the short-time Fourier transform, please see Smith (2011).

6. Experimental Results

To test the proposed method, a prototype user interface was built similar to Fig. 1 and tested on two sets of sound examples. For the first comparison, five mixture sounds of two sources each were tested. The original ground truth sources for each example were normalized to have a maximum of 0 dB gain and summed together to create the mixture sound. The mixture sounds were then separated using the proposed method over the course of five minutes each. The five mixture sounds include: ambulance siren + speech (S), cell phone ring + speech (C), drum + bass loop (D), orchestra + coughing (O), and piano chords + incorrect piano note (P). For a second comparison, four example rock/pop songs (S1, S2, S3, S4) from the Signal Separation Evaluation Campaign (SiSEC) database (SiSEC, 2011) were tested with the challenge of removing vocals from background music over the course of thirty minutes, similar to the evaluation of (Lefevre et al., 2012).

The results for both datasets were then compared against a baseline PLCA algorithm and an oracle algorithm. The baseline algorithm uses unsupervised PLCA with no training data or user-interaction to provide an approximate empirical lower bound on the results. The oracle algorithm uses the ground truth spectrogram data to compute the source probability masking filter $p(s|f,t)$ directly as the ratio of the ground truth source spectrogram divided by the mixture spectrogram to provide an approximate empirical upper bound on the results. In addition to the baseline and oracle results, the four rock/pop song results were compared against the method of Lefévre (2012) and Durrieu (2012), which, to our knowledge, are the only comparable methods that have some form of user-input and allow separation without training data.

Table 1. SDR, SIR, and SAR (in dB) for the first five example recordings using 100 dictionary elements/source.

Eval	Method	C	D	O	P	S
SDR	Oracle	26.9	15.1	12.2	26.1	26.7
	Baseline	-0.6	0.2	1.1	0.9	-4.1
	Proposed	24.8	11.0	9.7	22.0	21.8
SIR	Oracle	34.1	20.0	16.6	29.9	34.3
	Baseline	0.1	0.9	2.2	1.1	0.2
	Proposed	35.0	19.1	14.6	26.3	29.0
SAR	Oracle	27.9	16.8	14.6	28.8	27.6
	Baseline	14.0	12.6	10.5	17.5	7.0
	Proposed	25.8	12.6	11.7	24.3	23.2

For both test sets, the standard BSS-EVAL suite of metrics were used to evaluate performance (Vincent et al., 2006). The suite includes three separate metrics including the Source-to-Interference Ratio (SIR), Source-to-Artifacts Ratio (SAR), and Source-to-Distortion Ratio (SDR). The SIR measures the level of suppression of the unwanted sources, the SAR measures the level of artifacts introduced by the separation process, and the SDR gives an average measure of separation quality that considers both the suppression of the unwanted sources and level of artifacts introduced by the separation algorithm compared to ground truth. All three metrics have units of decibels (dB) and consider higher values to be better.

We illustrate two example sets of input and output spectrograms in Fig. 4 and display the complete evaluation results in Table 1 and 2. For both tests, a fixed number of basis vectors $N_z = 100 + 100$ were used. As shown, our proposed method outperforms the baseline, the method of Lefévre, and the method of Durrieu in all metrics for all examples. Note, the method of Durrieu previously ranked best SDR on average for the 2011 SiSEC evaluation campaign for removing vocals. In addition, in certain cases, the proposed method even performs near the quality of the ideal mask. Audio and video demonstrations can be found at `https://ccrma.stanford.edu/~njb/research/iss`.

Finally, to show how the proposed method behaves when varying the number of basis vectors per source, we performed separation for the first set of example sounds, then with the annotations fixed, varied the number of basis vectors and recomputed the results. Fig. 5 displays the SDR for the experiment, which shows that the method is relatively insensitive N_z, as long as the size is sufficiently large. This is notable in that the proposed method does not require the use of model selection to decide the number of basis vectors to use for a given separation task.

Table 2. SDR, SIR, and SAR (in dB) results for the four SiSEC rock/pop songs.

Eval	Method	S1	S2	S3	S4
SDR	Oracle	13.2	13.4	11.5	12.5
	Baseline	-0.8	0.2	-0.2	1.4
	Lefévre	7.0	5.0	3.8	5.0
	Durrieu	9.0	7.8	6.4	5.9
	Proposed	9.2	11.1	7.8	7.9
SIR	Oracle	17.8	18.0	17.5	19.5
	Baseline	0.5	1.6	0.9	3.1
	Lefévre	13.0	14.1	8.8	11.5
	Durrieu	16.4	16.8	13.0	12.6
	Proposed	17.4	20.1	14.8	13.8
SAR	Oracle	15.4	15.4	13.1	13.6
	Baseline	8.9	8.5	8.8	10.0
	Lefévre	8.9	7.3	6.1	6.5
	Durrieu	10.5	9.0	8.0	8.3
	Proposed	10.7	12.0	9.0	9.5

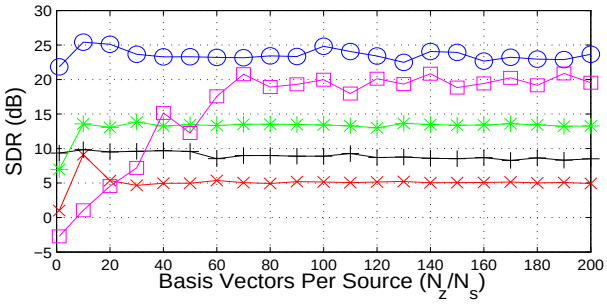


Figure 5. Comparison of SDR (in dB) to the number of basis vectors per source. Examples include Phone (blue, circle), Drum (red, x-mark), Orchestra (black, plus), Piano (green, star), and Siren (magenta, square).

7. Conclusions

To perform source separation when no isolated training data is available, we propose an interactive, weakly supervised separation technique. The method employs a user to interactively constrain a latent variable model by way of a new efficient posterior regularized EM algorithm. The use of PR allows for constraints that would be difficult to achieve using standard prior-based regularization and adds minimal additional computational complexity. A prototype user interface was developed for evaluation and tested on several example mixture sounds, showing the proposed method can achieve state-of-the-art results on real-world examples.

Acknowledgments

This work was performed, in part, while Nicholas J. Bryan was an intern at Adobe Research.

References

Benaroya, L., Donagh, L.M., Bimbot, F., and Gribonval, R. Non negative sparse representation for wiener based source separation with a single sensor. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, volume 6, april 2003.

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

Cohn, D., Caruana, R., and Mccallum, A. Semi-supervised clustering with user feedback. Technical report, 2003.

Durrieu, J.-L. and Thiran, J.-P. Musical audio source separation based on user-selected f0 track. In *The 10th International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, pp. 438–445, 2012.

Févotte, C., Bertin, N., and Durrieu, J.-L. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural Computation*, 21(3):793–830, March 2009.

Graça, J., Ganchev, K., and Taskar, B. Expectation maximization and posterior constraints. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

Graça, J., Ganchev, K., Taskar, B., and Pereira, F. C. N. Posterior vs parameter sparsity in latent variable models. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 664–672, 2009.

Hofmann, T. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pp. 50–57, New York, NY, USA, 1999. ACM.

Lee, D. D. and Seung, H. S. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 556–562. MIT Press, 2001.

Lefevre, A., Bach, F., and Févotte, C. Semi-supervised nmf with time-frequency annotations for single-channel source separation. In *In the Proceedings of The International Society for Music Information Retrieval (ISMIR) Conference*, 2012.

Raj, B. and Smaragdis, P. Latent variable decomposition of spectrograms for single channel speaker separation. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 17 – 20, oct. 2005.

SiSEC, 2011. Professionally produced music recordings. In *Signal Separation Evaluation Campaign (SiSEC)*, 2011. `http:`http://sisec.wiki.irisa.fr/tiki-index.php`//sisec.wiki.irisa.fr/tiki-index.php`.

Smaragdis, P. and Brown, J.C. Non-negative matrix factorization for polyphonic music transcription. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 177 – 180, oct. 2003.

Smaragdis, P. and Raj, B. Shift-Invariant Probabilistic Latent Component Analysis. *MERL Tech Report*, 2007.

Smaragdis, P., Raj, B., and Shashanka, M. A Probabilistic Latent Variable Model for Acoustic Modeling. In *Advances in Neural Information Processing Systems (NIPS), Workshop on Advances in Modeling for Acoustic Processing*, 2006.

Smaragdis, P., Raj, B., and Shashanka, M. Supervised and semi-supervised separation of sounds from single-channel mixtures. In *International Conference on Independent Component Analysis and Signal Separation*, pp. 414–421, Berlin, Heidelberg, 2007. Springer-Verlag.

Smith, J. O. *Spectral Audio Signal Processing*. `http:`http://ccrma.stanford.edu/ jos/sasp/`//-ccrma.stanford.edu/~jos/sasp/`, 2011. online book.

Vincent, E., Gribonval, R., and Fevotte, C. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462 –1469, july 2006.

Virtanen, T. Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuity and Sparseness Criteria. *IEEE Transactions on Audio, Speech and Language Processing (TASLP)*, 15(3):1066–1074, March 2007.