How To Find A Better Convex Hull

Abstract: Propose a nearest neighbor NMF algorithm for Unsupervised Learning (Blind Source Separation), which relaxes the assumption we used and gives reasonable good results.

Key words and phrases: Unsupervised Learning,Blind source separation, convex hull

1. Introduction The model for Unsupervised Learning (Blind Source Separation) can be expressed as

$$X = A * S + N \tag{1.1}$$

In this report, we will focus on the non-negative components separation.

$$X = A * S + N \quad S_{i,j} \geq 0, A_{i,j} \geq 0 \tag{1.2}$$

Here, X is the observed m-dimensional mixture signal, each row of X represents one mixture signal. A is mixing matrix. S represents n-dimensional source signals, $n <= m$, each row of S represents one source signal. N represents noise(often assumed to be while Gaussian). The goal here is to recover both A and S given only X. This is a ill-posed problem. Different assumptions are proposed based on application fields.

Two types of objective function are considered for blind source separation problem. one is

$$\text{minimize} \parallel X - A * S \parallel^2 \tag{1.3}$$

The other is Kullback-Leibler divergence cost function. In this report, we will focus on the first objective function. One view to understand all these different algorithms is to look at constraints and regularizers. For example, one side of the spectrum of these algorithms is kmeans algorithm.

$$\underset{S}{\text{argmin}} \quad \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2 \tag{1.4}$$

when basis A is fixed, the kmeans algorithm encoding corresponding to vector quantization. It gives strongest constraint that each column of S has only 1 one and others are 0s. We can reform Kmeans by the following:

$$\begin{aligned} &\min_{A,S} && \| X - A * S \|^2 \\ &\text{subject to} && \text{each column of } S \text{ has only 1 one and others are 0s.} \end{aligned} \tag{1.5}$$

where the columns for A are μ_i's .

At the other side of the spectrum for these algorithms , is Non-negative Matrix Factorization(NMF).

$$\begin{aligned} &\min_{A,S} && \| X - A * S \|^2 \\ &\text{subject to} && S_{i,j} \geq 0, A_{i,j} \geq 0 \end{aligned} \tag{1.6}$$

Although the function $\| X - A * S \|^2$ are convex in A only or S only, they are not convex in both variables together. Therefore it is unrealistic to expect an algorithm to solve this problem in the sense of finding a global minima. However, there are many techniques from numerical optimization that can be applied to find local minima. When we use the following update rule, S and A will converge to a local minimum. Gradient descent is perhaps the simplest technique to implement, but convergence can be slow. Other methods such as conjugate gradient have faster convergence, at least in the vicinity of local minima, but are more complicated to implement than gradient descent.The convergence of gradient based methods also have the disadvantage of being very sensitive to the choice of step size, which can be very inconvenient for large applications. One updates rules are the following:

$$\begin{aligned} S_{\alpha\mu} &\leftarrow S_{\alpha\mu} \frac{(A^T X)_{\alpha\mu}}{(A^T A X)_{\alpha\mu}} \\ A_{\beta\alpha} &\leftarrow A_{\beta\alpha} \frac{(X S^T)_{\beta\alpha}}{(A S S^T)_{\beta\alpha}} \end{aligned} \tag{2.1}$$

This is proved to converge.

In the middle of the spectrum, we have the algorithm proposed by Naanaa,

I call it self-convex-hull algorithm.

$$\begin{aligned} \min_{A,S} \quad & \| X - A * S \|^2 \\ \text{subject to} \quad & S_{i,j} \geq 0, A_{i,j} \geq 0 \\ \text{and} \quad & \text{S has columns of } \begin{bmatrix}1\\0\\0\\0\end{bmatrix} \quad \begin{bmatrix}0\\1\\0\\0\end{bmatrix} \quad \begin{bmatrix}0\\0\\1\\0\end{bmatrix} \quad \begin{bmatrix}0\\0\\0\\1\end{bmatrix} \end{aligned} \tag{2.2}$$

The way to compute A and S under this assumption is quite simple. Notice that the set of all the columns of mixing matrix A are a subset of the columns of X. Naanaa proposed the following algorithm.

1. Discard from X those columns X^j such that $\|X_j\| \leq \varepsilon$ Note X the resulting matrix.

2. Form the matrix X consisting of all the mutually nonlinear columns of X.

3. For each column X^j $(j = 1, \ldots, m.)$, solve the optimization problem:

$$\begin{aligned} \text{minimize} \quad & \left\| X^{\not\otimes j}\alpha(j) - X^j \right\|, \\ \text{subject to} \quad & \alpha_i(j) \geq 0 (j = 1, \ldots, m.) \end{aligned} \tag{2.3}$$

and compute score for X^j

$$score_j = \left\| X^{\not\otimes j}\alpha(j) - X^j \right\| \tag{2.4}$$

4. Select the n columns of X which are associated with the highest score and form matrix A

We can see that, comparing with Kmeans, instead of requiring all the columns have only 1 one and others are 0s, this algorithm only require existing the columns of

$$\begin{bmatrix}1\\0\\0\\0\end{bmatrix} \quad \begin{bmatrix}0\\1\\0\\0\end{bmatrix} \quad \begin{bmatrix}0\\0\\1\\0\end{bmatrix} \quad \begin{bmatrix}0\\0\\0\\1\end{bmatrix} \tag{2.5}$$

In this report , my goal is to relax the constraint even more. In real data, e.g. time series data, we might not be able to find a time point that all other source

signals totally disappear except one. But we might find several time points that only one signal is dominated . Mathematically speaking, my assumption is that there existing columns of

$$\begin{bmatrix} 1 \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} \quad \begin{bmatrix} \epsilon_1 \\ 1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} \quad \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ 1 \\ \epsilon_3 \end{bmatrix} \quad \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ 1 \end{bmatrix} \tag{2.6}$$

2. Motivation An important observation is that, geometrically thinking, we can consider the column vectors of matrix A as the support vectors for a convex hull. All the column vectors of X are our observed data points that are in such convex hull. Let A_i represents the columns of matrix A, or geometrically speaking, the support vectors. X_j are the column vectors of X, or geometrically speaking , the data vectors we can observe .The BSS problem is how to find the correct support vectors for the observed data points. To comparing all these algorithms, we would like to represent them in such geometric point of view. The Kmeans algorithm is saying all the column vectors of are actually support vector themselves. No point is actually inside the convex hull. The NMF algorithm is make sure all the data vectors X_j are in such convex hull, but put no constraints on where these support vectors A_i are. The self-convex-hull algorithm is saying the support vectors A_i are a subset of the data vectors of X_js. It means we pick the best support vectors from the given data vectors. Under my assumption that there existing columns of

$$\begin{bmatrix} 1 \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} \quad \begin{bmatrix} \epsilon_1 \\ 1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} \quad \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ 1 \\ \epsilon_3 \end{bmatrix} \quad \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ 1 \end{bmatrix} \tag{2.1}$$

I would like the support vector to form a slightly bigger convex hull, comparing with self-convex-hull algorithm.

Come back to the optimization point of view again.Again, let A_i represents the columns of matrix A, or geometrically speaking, the support vectors. X_j are the column vectors of X, or geometrically speaking , the data vectors we can observe . XN_{i1} is the corresponding nearest neighbor point of A_i in the data

vectors set. For self-convex-hull algorithm, since the support vectors are picked from the data vectors, which means the following optimization algorithm

$$\begin{aligned} \min_{A,S} \quad & \| X - A * S \|^2 + \lambda \| A_i - XN_{i1} \|^2 \\ \text{subject to} \quad & S_{i,j} \geq 0, A_{i,j} \geq 0, \lambda = +\infty \end{aligned} \tag{2.2}$$

$\lambda = +\infty$ here to force the support vector to be chosen from the data vector sets.

3. Our Approach Look at the optimization (2.2), comparing with my assumption that there existing several columns of

$$\begin{bmatrix} 1 \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} \quad \begin{bmatrix} \epsilon_1 \\ 1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} \quad \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ 1 \\ \epsilon_3 \end{bmatrix} \quad \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ 1 \end{bmatrix} \tag{3.1}$$

I came out of two ideas. The key here is notice that the assumption means several data vector points will locate near the support vectors. So we should penalize the distance from the support vector to its k-nearest neighbor points, which gives the following optimization problem

$$\begin{aligned} \min_{A,S} \quad & \| X - A * S \|^2 + \lambda \| A_i - 1/K * (XN_{i1} + XN_{i1} + \ldots + XN_{iK}) \|^2 \\ \text{subject to} \quad & S_{i,j} \geq 0, A_{i,j} \geq 0, \lambda \geq 0 \end{aligned} \tag{3.2}$$

λ here is not $+\infty$ but a positive parameter, adjusting how large I want the convex hull to be. When λ gets bigger, the convex hull gets smaller. Applying alternative direction method,we can optimize by alternating the following steps

Fixed S, update A

1.$A = ((SS^T)^{-1}SX^T)^T$

2.$A_i = A_i - \lambda * (A_i - 1/K * (XN_{i1} + XN_{i1} + \ldots + XN_{iK}))$

3.project all negative element of A to 0.

Fixed A, update S

1.$S = (A(A^TA)^{-1})^TX$

2.project all negative elements of S to 0.

4. Data Set and Experiment Results At this stage , I only use synthetic data to evaluate the performance of the algorithm. 3 source signals are considered here, since we can view the convex hull in dimension 3.

4. Future Work

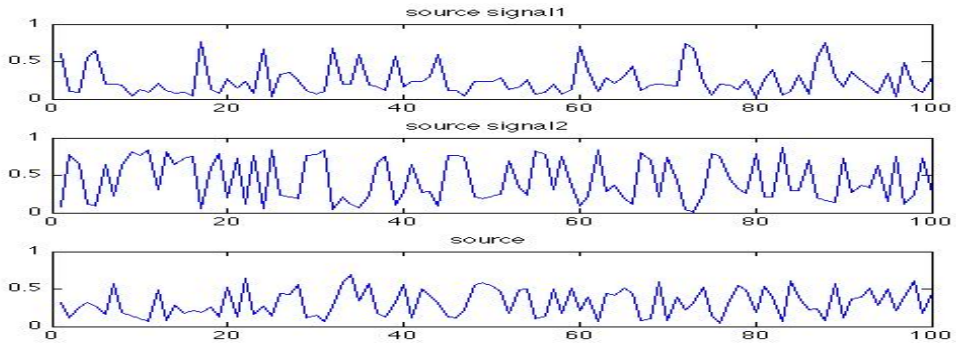

Figure 5.1: e.g.1 Source Signals

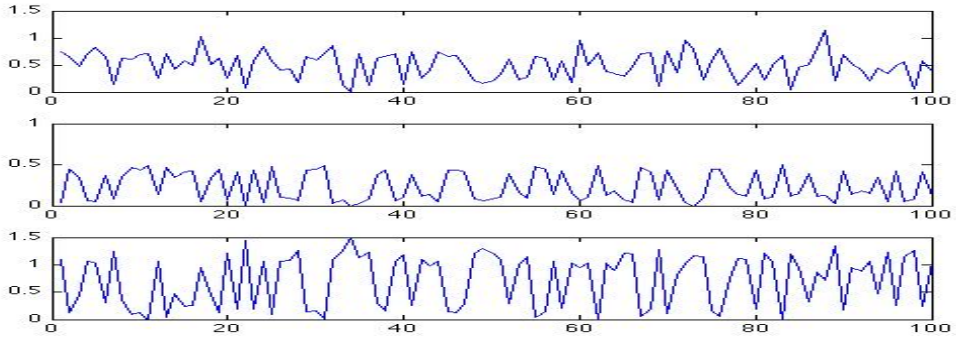

Figure 5.2: e.g.1 Recovered By NMF Signals

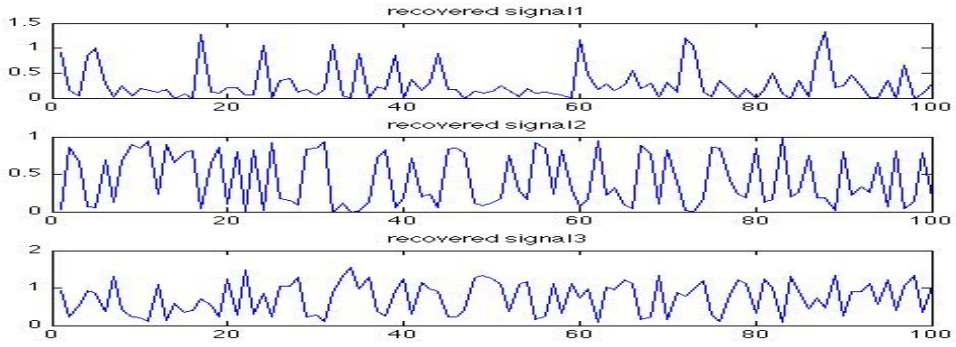

Figure 5.3: e.g.1 Recovered By New Algorithm Signals

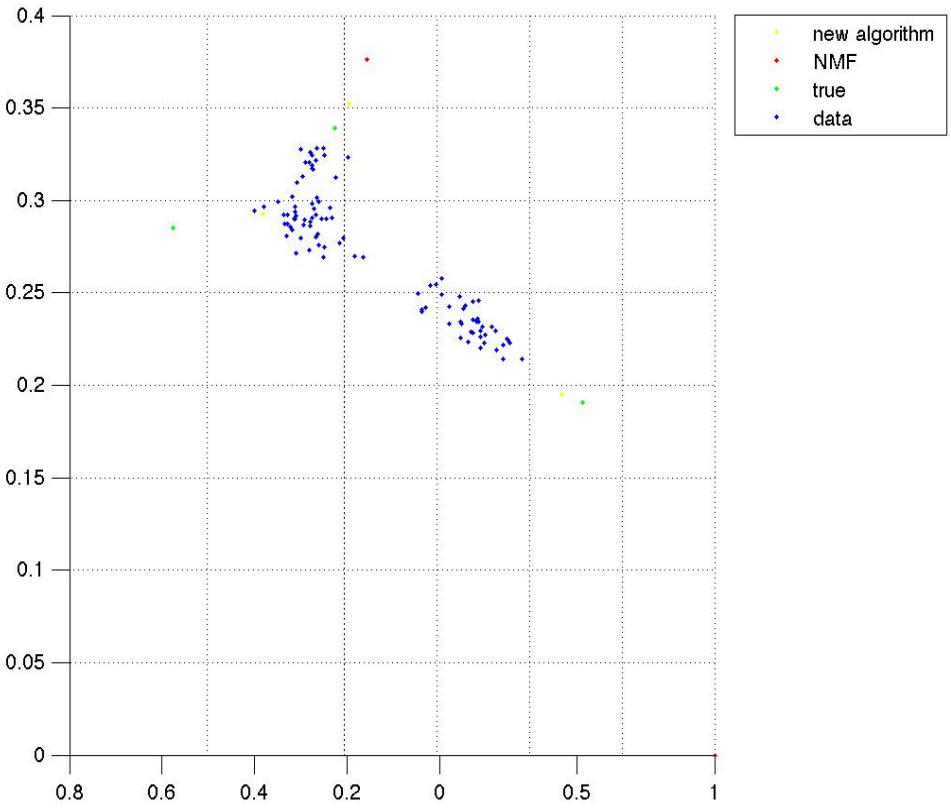

Figure 5.4: e.g.1 data vectors and support vectors visualization

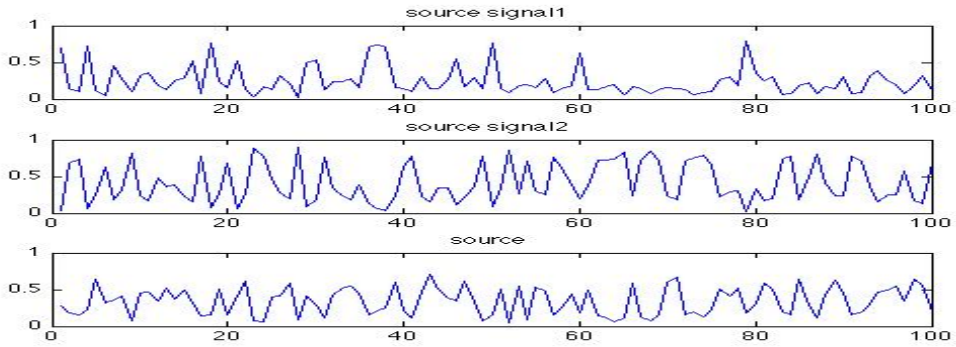

Figure 5.5: e.g.2 Source Signals

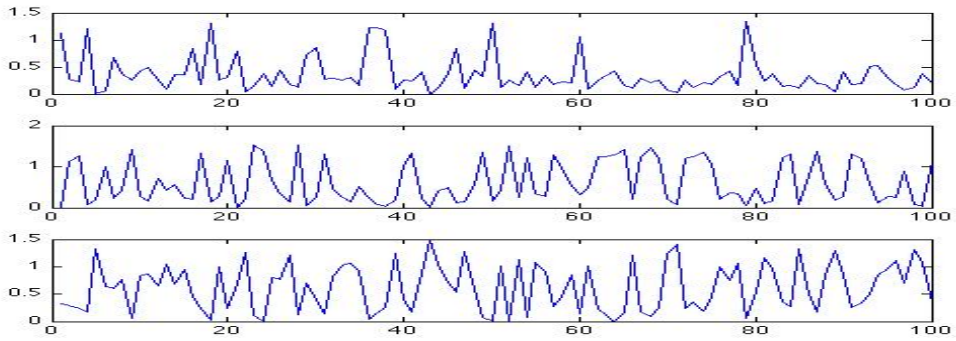

Figure 5.6: e.g.2 Recovered By NMF Signals

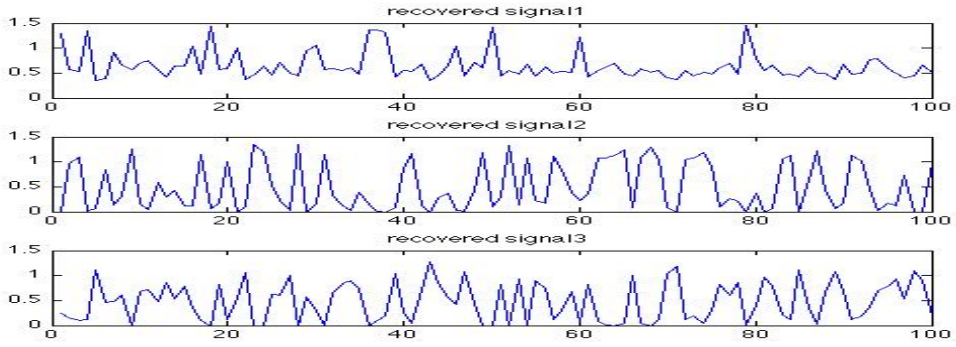

Figure 5.7: e.g.2 Recovered By New Algorithm Signals

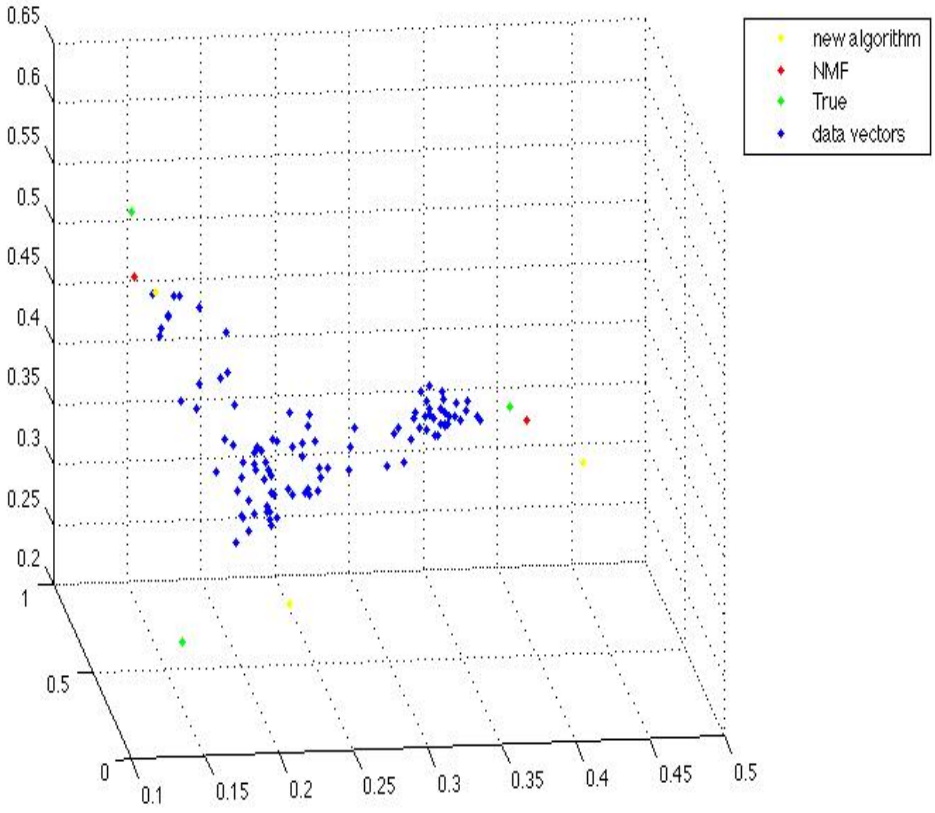

Figure 5.8: e.g.2 data vectors and support vectors visualization

Bibliography

first author affiliation
E-mail: (first author email)
second author affiliation
E-mail: (second author email)