# Vocal Extraction From Music Using RPCA Decomposition

**Authors**
Samuel Frank - smf2147
Shahriar Mokhtari-Sharghi - sm161
Joaquín Ruales - jar2262
Nicholas Ursa - nju2012

## Abstract

We explore the application of Principal Component Analysis for extracting melody and vocals from a piece of music. In order to solve this problem, we explore Robust Principal Component Analysis as a technique for making PCA robust to large sparse noise, and we investigate multiple techniques for solving the RPCA problem. We find that by using Augmented Lagrangians and the ADMIP methods, we are able to efficiently separate out vocals from a piece of music.

## 1 Introduction

In recent years, Principal Component Analysis has become a popular technique for both reducing the dimensionality of data and for separating out noise from low-rank matrices. However, while PCA is quite good at separating out dense low-level noise, it is not robust to arbitrarily large noise, no matter how sparse that noise is.

One context in which large, sparse noise is relevant is in the extraction of vocals from music. In a piece of music, one can often consider the background music to be low-rank and the vocals to be sparse noise. We explore techniques for making PCA robust to this large, sparse noise and for using PCA to separate out the vocals from the background in a piece of music.

## 2 Problem setting

### 2.1 STFT transform

Our goal is to be able to separate a vocal melody from an instrumental background in a piece of music. We formulate this problem by moving our linear signal into the time and frequency domain via a short-time Fourier Transform (STFT). This is done by partitioning the original signal into overlapping chunks in time, and then doing a Fourier transform on a regular series of frequencies. 100Hz, 200 Hz, etc. There is some overlap of each frequency bin as well, known as a Hamming window, to approximate a flat frequency response.

A Fourier transform yields 2 components: $sin(x)+cos(y)$. However we can also characterize this as an amplitude and angle by taking the norm and angle of the resulting 2D vector (x,y), and turning it into a complex number: $a+ic$. In general, audio applications work with the magnitude vector alone, discarding the complex component temporarily. This yields a series of spectral vectors, forming a matrix, $D$. After transformations to $D$ are carried out we need a complex component again to the the inverse transform to make the original signal. Typically the originals are added back (a kind of cheat) however there are technies to infer them from context.

Our modeling assumption is that in this form, vocal melodies would form a matrix that is mostly empty, and has relatively high rank, because a singer tends to modulate a lot in pitch and harmonics as different words are pronounced and they glissando from note to note. The background instruments, however, tend to repeat similar sets of spectra as they stay on a harmony for sometimes bars at a time. Since the typical song only consists of 5 or 6 chords, the whole rhythm section can be characterized with linear combinations of relatively few spectra, leading to a low rank approximation. In addition, the rhythm section tends to create a more solid wall of frequencies.

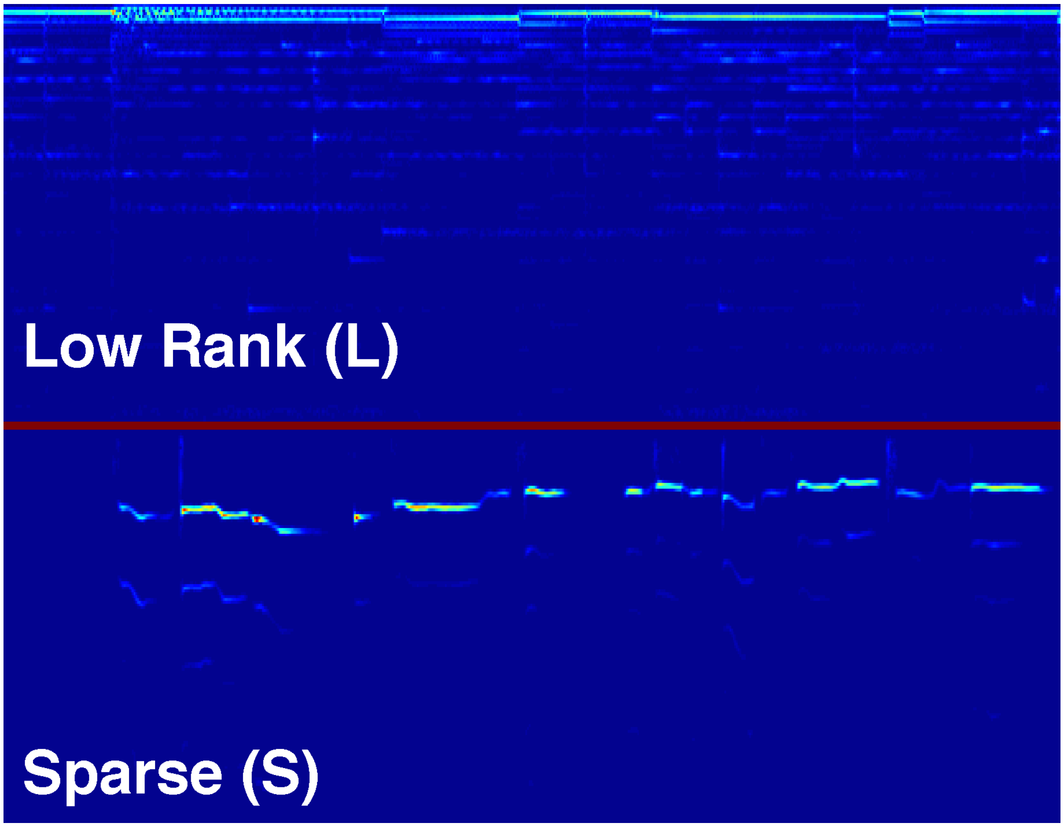


Figure 1: Background music shows low rank structure ($L$), a vocal melody shows sparse structure ($S$)

A matrix decomposition which takes a matrix D and separates it into the sum of a low-rank matrix and a sparse matrix ($L + S$) as above, should then keep the melody in $S$ and the background music in $L$.

There are many other application areas that follow a similar structure and can be cast as the same decomposition. For example, we can take a 2D image and make a 1D vector of pixel values out of it by stacking. Then we can take security camera footage and take successive frames as different rows. The background is static and low rank. Any movement will appear as a sparse matrix.

Other use cases include [7]:

- Facial recognition where the same face is low rank and differing occlusions are sparse
- Matrix rigidity (minimum number of entries you need to change to make a matrix low rank = $|S|$)
- Model selection

Naively, we might want to separate out the low-rank matrix L using principal component analysis. However, PCA is not robust to outliers and large, sparse noise.

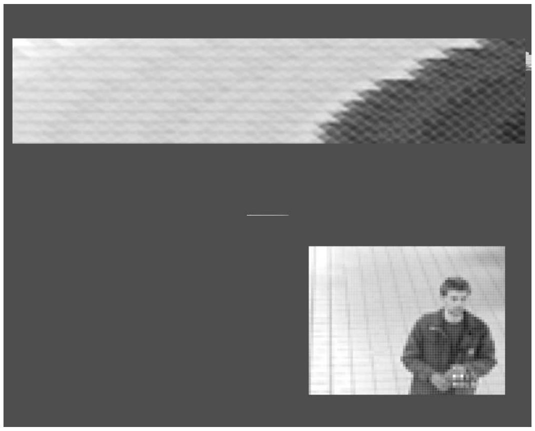


Figure 2: Casting motion detection as $L$ + $S$

## 3 Why PCA is not Robust to Outliers

Assume that our data is given by matrix $D = [D_1 \ D_2 \ \ldots \ D_n]$, where each $D_i$ is one data column vector. PCA uncovers the principal directions in a data set by transforming the data into a special orthogonal coordinate system. In this coordinate system, the data obtains the most variance when projected onto the first axis, the second highest variance when projected onto the second axis, and so on. Assuming out data to be mean-centered, the first component (the first axis in our new coordinate system) is given by the optimization problem

$$\arg \max_{\|w\|=1} \sum_{i=1}^{n} (D_i \cdot w)^2,$$

or equivalently,

$$\arg \max_{\|w\|=1} (w^T (DD^T) w),$$

Where $DD^T$ is the covariance matrix of the data.

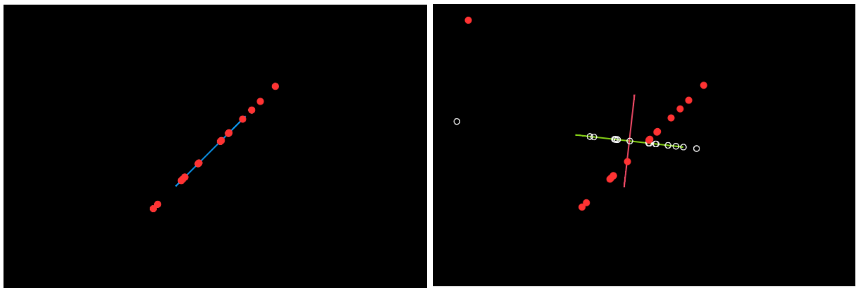
Recall that any variance statistic—much like any mean statistic—is prone to being sensitive to outliers. Because PCA has as its objective to maximize variance, it is possible to trick PCA into believing that its least significant component is the most significant. The only thing we need in order to achieve this is to add a single gross outlier in the direction of this least component. To exemplify, we have created a visualization that allows a user to move data points in two dimensions and see the principal components returned by PCA in real time. In figures 3.a and 3.b, you can see the result of PCA applied to linear data, versus PCA applied to data with one outlier. The red circles indicate the data points, the line segments indicate the principal components scaled according to the standard deviation in that direction, and the white circles indicate the position of the data points after being projected onto the first component.

## 4 Robust PCA

Robust PCA avoids the aforementioned outlier-sensitivity of Principal Component Analysis by taking noise into account in the optimization model. Instead of concerning itself with maximizing variance, Robust PCA directly follows our objective of separating the data into the sum of a low-rank component, $L$, and a sparse component, $S$.

The low-rank component ensures that the principal component coordinate system is low-dimensional. The sparse component, on the other hand, ensures that we only get rid of sparse outliers and don't discard important dense information. One of the seminal papers on RPCA [4, Wright et al. 2009] describes the optimization problem as

$$\min_{L \in \mathbb{R}^{\gtrdot \times \ltimes}} rank(L) + \xi \|D - L\|_0$$



(a) Results of PCA on low-rank data with no outliers.

(b) Results of PCA on low-rank data with one large outlier.

Figure 3: Traditional PCA

for some suitable parameter $\xi$ but it also explains that this problem is in fact intractable and highly nonconvex. The paper then introduces the Principal Component Pursuit (PCP) as a tractable, convex approximation to the above formulation that uses the nuclear norm instead of the rank, and the 1-norm instead of the zero norm.

$$\min_{L\in\mathbb{R}^{\gtrdot\times\ltimes}} \|L\|_* + \xi\|D - L\|_1$$

Where $*$ denotes the nuclear norm, that is, the sum of the singular values of the matrix. Below, we review several optimizations and improvements on the Principal Component Pursuit, and we have coded an interactive MATLAB example in order to compare the robustness of RPCA to that of PCA - a screenshot is shown in Figure 5.

## 5 Solving RPCA using the Augmented Lagrangian Method

Because the Principal Component Pursuit problem is convex, we can solve it using the Augmented Lagrangian Method, which is the approach that Huang et. al. take [1]. The augmented Lagrangian problem solves a Lagrangian to which an extra term has been added containing the Frobenius norm of $D - L - S$, i.e. the Frobenius norm of the error between the original data and the sum of the low rank and sparse matrices. Lin [2] gives the Lagrangian function as:

$$\mathcal{L}(L, S, Y, \mu) = ||L||_* + \lambda||S||_1 + < Y, D - L - S > + \frac{\mu}{2}||D - L - S||_F^2$$

In this expression, $\mu$ is a weight to the Frobenius norm: the larger the value of $\mu$, the more weight is put on ensuring that the low rank and sparse matrices sum to the original matrix. $L$ represents the low-rank matrix and $S$ represents the sparse, noisy matrix. $D$ represents the original data matrix.

There are several algorithms to solve the augmented Lagrangian; Huang et. al. suggest using the inexact ALM method for computational efficiency. The iterative algorithm is as follows. At each iteration $k + 1$:

1. Solve for the value of $L_{k+1}$ that minimizes $\mathcal{L}(L, S_k, Y_k, \mu_k)$
2. Solve for the value of $S_{k+1}$ that minimizes $\mathcal{L}(L_{k+1}, S, Y_k, \mu_k)$
3. Set $Y_{k+1} = Y_k + \mu_k(D - L_{k+1} - S_{k+1}$
4. Set $\mu_{k+1} = \rho \cdot \mu_k$ for a fixed $\rho$

[2]

This algorithm increases $\mu$ as a geometric sequence, multiplying it by a constant factor at each iteration. The result of this is that this algorithm essentially operates in two phases. In the first phase, the algorithm seeks to minimize the rank of $L$ and the sparsity of $S$, and in the second phase the algorithm starts at a good solution and adjusts the matrices so that $L + S = D$. We implemented

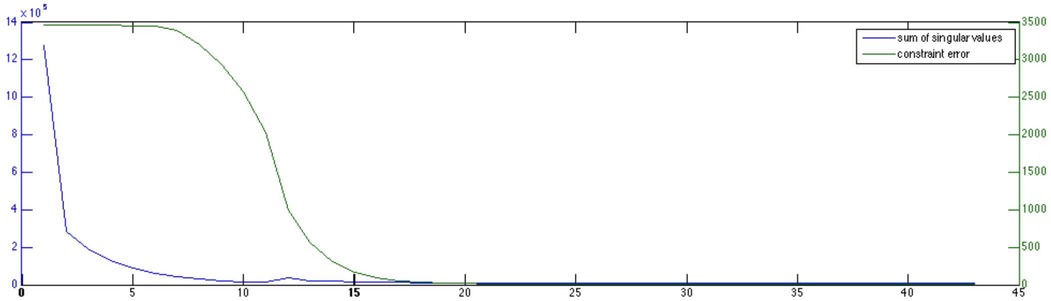
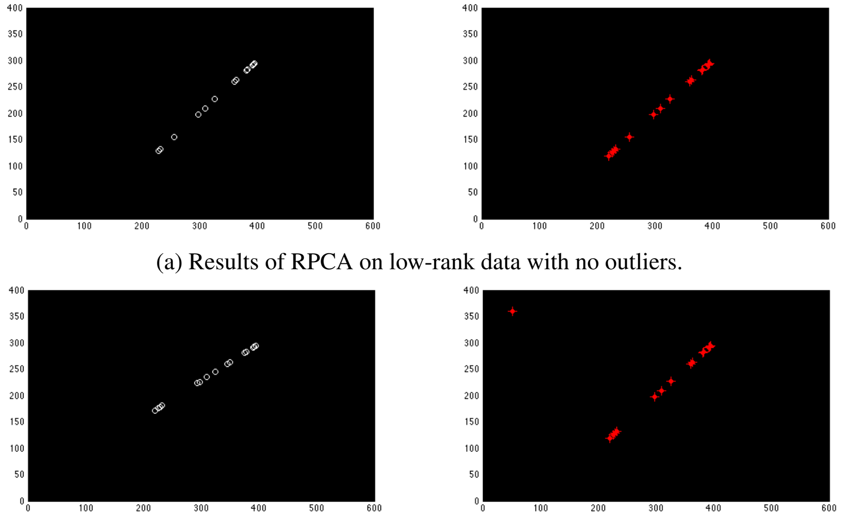


Figure 4: Inexact ALMM method applied to an artificial low rank + sparse matrix..

this algorithm on artificially constructed low-rank + sparse matrices; the results are below in Figure 4. Note that the error term for the low-rank/sparse component drops before the error term for the $(D - L - S)$ component.

We also modified the traditional PCA demo to show the effects of RPCA on the same type of data. The results are below.



(a) Results of RPCA on low-rank data with no outliers.

(b) Results of PCA on low-rank data with one large outlier.

Figure 5: Robust PCA

# 6 Alternating direction method of multipliers with an increasing penalty sequence (ADMIP)

## 6.1 Stable principal component pursuit problem and known results

This section provides an exposition to a recent developed algorithm for solving *stable principal component pursuit* (SPCP) problem. The main referenced paper for the algorithm here is [12]. The algorithm for solving SPCP is a modification of the alternating direction method of multipliers (ADMM) where we use an increasing sequence of penalty parameters instead of a fixed penalty. The algorithm is based on partial variable splitting and works directly with the non-smooth objective function.

Suppose a matrix $D \in \mathbb{R}^{m \times n}$ is of the form $D = L0 + S0$, where $L0$ is a low-rank matrix, i.e. $rank(L0) \ll min\{m, n\}$, and $S0$ is a sparse matrix. The matrix $S0$ is interpreted as gross errors in the measurement of the low rank matrix $L0$. Wright et al. [4], CandLes et al. [5] and Chandrasekaran et al. [6] proposed recovering the low-rank L0 and sparse S0 by solving the principal component pursuit (PCP)

$$\min_{L\in\mathbb{R}^{m\times n}} \|L\|_* + \xi\|D-L\|_1 \tag{1}$$

where $\xi = \frac{1}{\sqrt{\max\{m,n\}}}$. Here the nuclear norm $\|L\|_* := \sum_{i=1}^r \sigma_i(L)$, where $\{\sigma_i(L)\}_{i=1}^r$ denotes the singular values of $L \in \mathbb{R}^{m\times n}$ and the $\ell_1$ norm $\|L\|_1 := \sum_{i=1}^m \sum_{j=1}^n |L_{ij}|$.

**Theorem 6.1.** *[5] Suppose $D = L^0 + S^0 \in \mathbb{R}^{m\times n}$. Let $r = rank(L^0)$ and $L^0 = U\Sigma V^T = \sum_{i=1}^r \sigma_i u_i v_i^T$ denote the singular value decomposition (SVD) of $L^0$. Suppose there exists $\mu > 0$ such that*

$$\max_i \|U^T e_i\|_2^2 \le \frac{\mu r}{m},\ \max_i \|V^T e_i\|_2^2 \le \frac{\mu r}{n},\ \|UV^T\|_\infty \le \sqrt{\frac{\mu r}{mn}}, \tag{2}$$

*where $e_i$ denotes the $i$-th unit vector, and the non-zero components of the sparse matrix $S_0$ are chosen uniformly at random. Then there exist constants $c$, $\rho_r$, and $\rho_s$, such that the solution of the PCP problem (1) exactly recovers $L^0$ and $S^0$ with the probability of at least $1 - cn^{-10}$, provided*

$$rank(L^0) \le \rho_r m\mu^{-1}(\log(n))^{-2} \text{ and } \|S^0\|_0 \le \rho_s mn, \tag{3}$$

*where the $\ell_0$-norm $\|S^0\|_0$ denotes the number of non-zero components of the matrix $S^0$.*

Now, suppose that the data matrix, $N^0$ is of the form $D = L^0 + S^0 + N^0$ such that $L^0$ is a low-rank matrix, $S^0$ is a sparse gross "error" matrix, $N^0$ is a dense noise matrix with $\|N^0\|_F \le \delta$, where the Frobenius norm $\|Z\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n Z_{ij}^2}$. In [8], it was shown that it was still possible to recover the low-rank and sparse components $(L^0, S^0)$ of $D$ by solving the *stable principal component pursuit* (SPCP) problem

$$\min_{L,S\in\mathbb{R}^{m\times n}} \{\|L\|_* + \xi\|S\|_1 : \|L + S - D\|_F \le \delta\}. \tag{4}$$

**Theorem 6.2.** *[8] Suppose $D = L^0 + S^0 + N^0$, where $L^0 \in \mathbb{R}^{m\times n}$ with $m < n$ satisfies (2) for some $\mu > 0$, and the non-zero components of the sparse matrix $S^0$ are chosen uniformly at random. Suppose $L^0$ and $S^0$ satisfy ( 3). Then for any $N^0$ such that $\|N^0\|_F \le \delta$, the solution $(L^*, S^*)$ to the SPCP problem (4) satisfies $\|L^* - L^0\|_F^2 + \|S^* - S^0\|_F^2 \le Cmn\delta^2$ for some constant $C$ with high probability.*

In some applications, some of the entries of $D$ in (4) may not be available. Let $\Omega \subset \{i : 1 \le i \le m\} \times \{j : 1 \le j \le n\}$ be the index set of the observable entries of $D$. Define the the projection operator $\pi_\Omega : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ as follows:

$$(\pi_\Omega(L))_{ij} = \begin{cases} L_{ij}, & \text{if}(i,j) \in \Omega \\ 0, & \text{otherwise} \end{cases}$$

For applications with missing observation, Tao and Yuan [9] proposed recovering the low rank and sparse components of $D$ by solving

$$\min_{L,S\in\mathbb{R}^{m\times n}} \{\|L\|_* + \xi\|S\|_1 : \|\pi_\Omega(L + S - D)\|_F \le \delta\}. \tag{5}$$

### 6.2 ADMIP: Alternating Direction Method with Increasing Penalty Algorithm

Suppose

$$\chi = \{(Z,S) \in \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} : \|\pi_\Omega(Z + S - D)\|_F \le \delta\}$$

is the feasible set in 5 and $1_\chi(.,.)$ denote the indicator function of the closed convex set $\chi \subset \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n}$. We split $L$ variable in (4) to get the following equivalent problem:

$$\min_{L,Z,S\in\mathbb{R}^{m\times n}} \{\|L\|_* + \xi\|S\|_1 + 1_\chi(Z,S) : L = Z\} \tag{6}$$

The augmented Lagrangian function of (6) is defined:

$$La_\rho = \{L, Z, S; Y) = \|L\|_* + \xi\|S\|_1 + 1_\xi(Z,S) + \langle Y, L - Z\rangle + \frac{\rho}{2}\|L - Z\|_F^2 \tag{7}$$

In each iteration of ADMIP we try to minimize (7). Below is the description for ADMIP algorithm for given $Z_0$, $Y_0$ and $\{\rho_k\}$

```
Require: $Z_0 \in \mathbb{R}^{m\times n}, Y_0 \in \mathbb{R}^{m\times n}$ $k \leftarrow 0$
  while $k \geq 0$ do
    $L_{k+1} \leftarrow \text{argmin}_L\{\|L\|_* + \langle Y_K, L - Z_k\rangle + \frac{\rho_k}{2}\|Z - L_{k+1}\|_F^2\}$
    $(Z_{k+1}, S_{k+1}) \leftarrow \text{argmin}_{\{(Z,S):\|\pi_\Omega(Z+S-D)\|_F\leq\delta\}}\{\xi\|S\|_1 + \langle -Y_k, Z - L_{k+1}\rangle$
    + $\frac{\rho_k}{2}\|Z - L_{k+1}\|_F^2\}$
    $Y_{k+1} \leftarrow Y_k + \rho_k(L_{k+1} - Z_{k+1})$
    $k \leftarrow k + 1$
  end while
```

It can be shown that in case that $\sum_k \frac{1}{\rho_k} < \infty$ the above algorithm converges. It has performed better than ADMM for some test cases.

## 7 Results

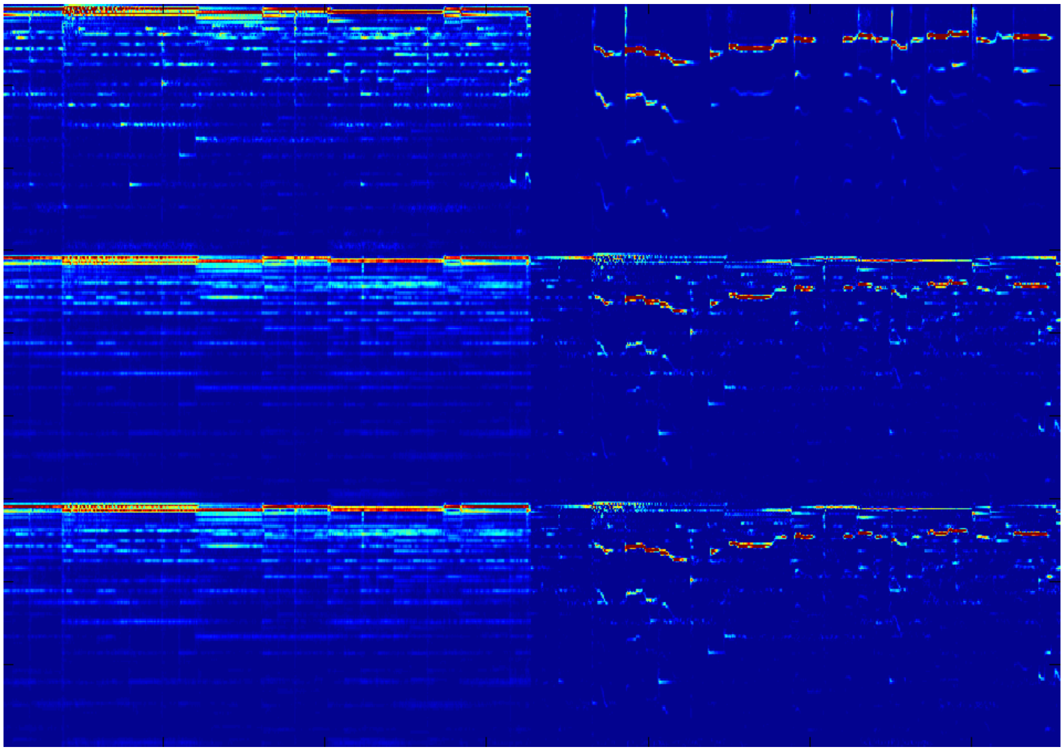


Figure 6: Both methods produced similar results.

We had examples of unmixed karaoke songs where we were able to start with seperate A (background) and E (melody) matrices. Then we combined them and had the algorithms try to separate them. We used the frobenius norm between the estimates and the actuals, divided by the norm of the actuals to have an approximation of signal to noise.

Table 1: Results (lower is better)

| Test | Metric | ALMM | ADMIP |
|---|---|---|---|
| Melody accuracy | $\|\hat{S} - S\|_{fro}/\|S\|_{fro}$ | 1.8464 | 1.8822 |
| Background accuracy | $\|\hat{L} - L\|_{fro}/\|L\|_{fro}$ | 1.5390 | 1.5168 |

The differences in accuracy between the two techniques was negligable, and could not be heard in the resulting audio.

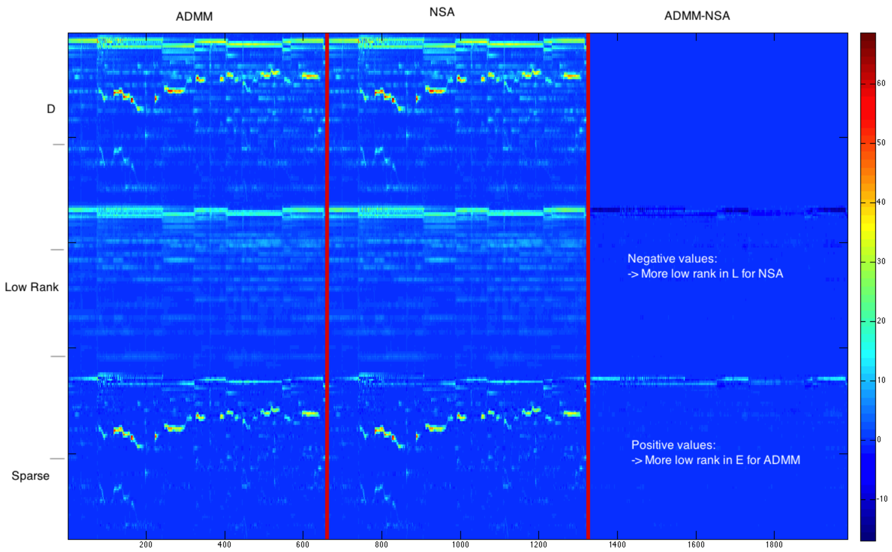


Figure 7: ALM algorithm vs. the ADMIP algorithm.

## 8 Conclusion

RPCA is a promising technique for removing vocals and melody from a piece of music. Using the principal component pursuit technique, RPCA can be reduced to a tractable problem. ALMM was a bit faster to converge, but ADMIP had a hyperparameter which may allow it to adapt to different audio settings.

## References

[1] Huang, Po-Sen, et al. ”Singing-voice separation from monaural recordings using robust principal component analysis.” Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on. IEEE, 2012.

[2] Lin, Zhouchen, Minming Chen, and Yi Ma. ”The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices.” arXiv preprint arXiv:1009.5055 (2010).

[3] N. S. AYBAT, G. IYENGAR Alternating direction method with increasing penalty for stable principal component pursuit. http://arxiv.org/abs/1309.6553

[4] J. Wright, Y. Peng, Y. Ma, A. Ganesh, and S. Rao *Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization,* in Proceedings of Neural Information Processing Systems (NIPS), December 2009.

[5] E. J. Cand‘es, X. Li, Y. Ma, and Wright J. *Robust principle component analysis?,* Journal of ACM, 58 (2011), pp. 137

[6] V. Chandrasekaran, S. Sanghavi, P. Parrilo, and A. Willsky, *Rank-sparsity incoherence for matrix decomposition,* SIAM Journal on Optimization, 21 (2011), pp. 572-596.

[7] Chandrasekaran, Venkat, et al. ”Sparse and low-rank matrix decompositions.” Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on. IEEE, 2009.

[8] Z. Zhou, X. Li, J. Wright, E. Cand‘es, and Y. Ma, *Stable principle component pursuit,* Proceedings of International Symposium on Information Theory, (2010).

[9] M. Tao and X. Yuan, *Recovering low-rank and sparse components of matrices from incomplete and noisy observations,* Proceedings of International SIAM Journal on Optimization, 21 (2011), pp. 57-81.

[10] Wright, John, and Yi Ma. ”Dense error correction via-minimization.” Information Theory, IEEE Transactions on 56.7 (2010): 3540-3560.

[11] N. S. Aybat. [2011] ”First Order Methods for Large-Scale Sparse Optimization.” Ph.D Thesis. Columbia University, United States.

[12] Richtarik, Takac, Ahipasaoglu. Alternating Maximization: Unifying Framework for 8 Sparse PCA Formulations and Efficient Parallel Codes. http://arxiv.org/abs/1309.6553

[13] Aybat, Necdet Serhat, Donald Goldfarb, and Shiqian Ma. ”Efficient algorithms for robust and stable principal component pursuit problems.” Computational Optimization and Applications (2012): 1-29.

[14] Xu, Huan, Constantine Caramanis, and Sujay Sanghavi. ”Robust PCA via Outlier Pursuit.” IEEE Transactions on Information Theory 58.5 (2012): 3047-3064.