Separation of Speech Sources in Under-Determined Case Using SCA and Time-Frequency Methods

Rahil Mahdian[*], Massoud Babaiezadeh[†], Christian Jutten[§]
[*†]Dept. of Electrical Engineering
Sharif University of Technology, Tehran, IRAN
[*] Email: r.mahdiyan@gmail.com
[†] Email: mbzadeh@sharif.edu
[§] Labratoire des Images et de signaux (LIS), Institut National Polytechnique de Grenoble (INPG), Paris, FRANCE
[§] Email: Christian.Jutten@inpg.fr

Abstract-**This paper presents a new algorithm for Blind Source Separation (BSS) of Instantaneous speech mixtures in under-determined case. A demixing algorithm which exploits the sparsity of speech signals in the short time Fourier transform (STFT) domain is proposed. This algorithm combines the modified k-means clustering procedure involved in the Line Orientation Separation Technique (LOST) with Smoothed l^0-norm minimization (SL0) method. First procedure along with a transformation into a sparse domain tries to estimate the mixing matrix, and the second method tries to extract the sources from the mixtures. Simulation results are presented and compared to the Degenerate Unmixing Estimation Technique (DUET) and LOST algorithms. It is shown in this article that improvements are achieved in two cases. One is the quality of source extraction when the number of mixtures is increased, and second is the speed of source separation when compared to the LOST algorithm. This speed enhancement is about 3 to 10 times comparing to the LOST algorithm. We called the proposed algorithm SL0-LOST.**

I. INTRODUCTION

Blind source separation problem consists of estimating N sources from M mixtures that are unknown functions of the sources. The noise free linear model for each sample is:

$$\mathrm{A}\mathbf{s} = \mathbf{x}. \tag{1}$$

Where $\mathbf{s} \in \mathbb{R}^N$ is the source random vector, $\mathbf{x} \in \mathbb{R}^M$ is the measurement random vector, and $\mathrm{A} \in \mathbb{R}^{M\times N}$ is the unknown mixing matrix. In under-determined case, when less measurements than sources are available (M<N), there is no unique inverse of the mixing matrix. Assumptions can be made about the nature of the sources. Such assumptions form the basis for most source separation algorithms. One powerful assumption is that the sources have a sparse representation. These methods have come to be known as "Sparse Methods" [1]. In many cases, even if the sources do not satisfy the sparsity premise, it is possible to apply a suitable linear transform like STFT, DCT, wavelet, etc. that does allow representing the signals in a new space in which the coefficients are sparse, as in [2].

Speech signals have nearly sparse representations in some transformation domains like STFT [1]. There are some basic algorithms to extract speech signals from under-determined linear systems of equations. Some of them exploit the sparsity of speech signals in frequency (STFT) domain. DUET is one of these basic algorithms and uses masking technique to extract sources from only 2 mixtures [3]. DUET has some extensions such as TIFROM and DESPRIT [4], [5]. These algorithms exploit the shortcomings of DUET like the WDO assumption, fixed number of mixtures and the method which DUET uses to estimate delay and attenuation parameters of the sources [6]. Another basic algorithm is LOST which exploits the sparsity of the scatter plot of speech signals in time-frequency domain [7], [8]. We will proceed with this algorithm in the next section. In this paper, a new BSS technique for extracting speech sources in an under-determined instantaneous environment is proposed. Thus, a two stage approach is adopted. In the first stage the mixing parameters are estimated by modified k-means clustering procedure combined with a transformation of mixtures into a sparse domain. These parameters are then used in the second stage to extract the sources. In particular, Smoothed l^0-norm minimization algorithm is used to estimate the sources in STFT domain [9]. Accordingly, the novel aspects of the proposed approach are as follows:

- We propose a two stage algorithm that unlike DUET [6] can make use of all available mixtures, both in mixing model estimation and in source extraction stage. Moreover, unlike DESPRIT [5], we don't need to impose a predetermined structure on the sensor array.
- We propose the use of SL0 technique which is shown [9] to significantly improve the total separation **performance** when compared to DUET and LOST algorithms. Moreover, it improves the separation **speed** when compared to LOST algorithm.
- We analyze the superficial capability of SL0 algorithm while involved in speech separation problems and extracting the remarkable **issues** during the use of this method for sparse separation of **speech sources**.

II. REVIEW OF LOST ALGORITHM

This algorithm treats clustering of data points into one-dimensional subspaces that cross the origin [7]. On the other hand, the subspaces correspond directly to columns of the mixing matrix [7]. So if the lines can be estimated from the data then an estimate of the mixing matrix can be trivially constructed. The LOST algorithm identifies the subspaces using a modified k-means clustering procedure. The cluster centers are now line orientations and the distance from cluster centers is now the distance from a line. Two types of LOST algorithms exist named hard-LOST and soft-LOST. In hard-LOST the k-means technique uses hard decisions, i.e. a data point is assigned to one cluster centre and no other [7]. In soft-LOST the Expectation Maximization (EM) algorithm is modified to search for line orientations [8]. This algorithm uses two stages, the mixing matrix is estimated using the previously mentioned modified k-means method combined with a transformation into a sparse domain (here STFT) at the first stage. In the second stage sources are extracted from the mixtures using ℓ^1-Norm optimization. A summary of the LOST algorithm is as follows [7], [8]:

Step1. A number of line orientations are randomly initialized.

Step2. Each point is assigned to a line. Hard-LOST assigns points to the closest line. Soft-LOST assigns points in a soft way to each of the lines, i.e. points near line i will be given a high weight when associated with cluster i and points far from that line are vice versa.

Step3. New line orientations are calculated based on the assignments obtained in step2.

Step4. Step2 and 3 are repeated until the line orientations converge.

III. REVIEW OF SL0 ALGORITHM

As it's stated in [9] SL0 is essentially an algorithm for obtaining sufficiently sparse solutions of under-determined systems of linear equations. To obtain the sparsest solution of A**s**=**x**, we may search for a solution of it having min-ℓ^0-norm; However searching for min-ℓ^0-norm is an intractable problem when the dimension increases ([9]). The min-ℓ^1-norm solution is the same as min-ℓ^0-norm and using fast Linear Programming (LP) algorithms, like Interior Point LP solvers, large scale problems become tractable but it's very slow. Matching Pursuit is another approach but it doesn't provide good estimation of the sources [9].

SL0 is a fast method for finding the sparse solution of an underdetermined system of linear equations. The main idea behind SL0 is to approximate the discontinuous function of ℓ^0 norm by a suitable continuous one, and minimizing it by means of a minimization algorithm which is used for continuous functions e.g., steepest descent method. It's shown in [9] that we can use an approximation to ℓ^0 -norm of a vector **s** ($\mathbf{s}=[s_1 \ldots s_n]^{\mathrm{T}}$) like this:

$$\|\mathbf{s}\|_0 \approx n - F_\sigma(\mathbf{s}) \quad (2)$$

Where $$F_\sigma(\mathbf{s}) = \sum_{i=1}^{n} \exp(-s_i^2 / 2\sigma^2) \quad (3)$$

Now, minimization of ℓ^0–norm is equivalent to maximization of F_σ for a sufficient small σ. F_σ can also be seen as a measure of sparsity of a vector (especially for small values of σ). Moreover, it's shown in [9] that this algorithm is about two or three orders of magnitude faster than the best interior point LP solvers, while providing the same accuracy.

IV. THE PROPOSED ALGORITHM

In this article a two stage policy is adopted for the new algorithm which we call it SL0-LOST. The block diagram of our algorithm is as follows:

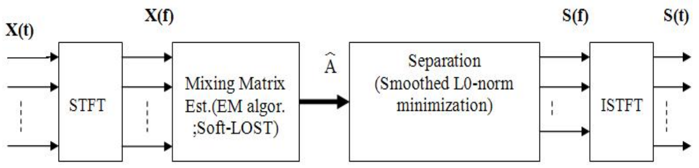
X(t)
X(f)
S(f)
S(t)
STFT
Mixing Matrix Est.(EM algor. ;Soft-LOST)
Â
Separation (Smoothed L0-norm minimization)
ISTFT

Figure1. Block Diagram of the proposed Algorithm

The proposed algorithm is as follows:

a) Transfer the x=[x(1),…,x(M)] to the STFT domain and create a scatter plot of it. Perform kurtosis scaling of the coefficients as mentioned in [7]. Then the transformed observations are plotted against each other.

b) Randomly initialize the N line orientation vectors $\mathbf{v_i}$, and initialize γ to a sufficient large value.

c) Partially assign each observation, x(t), to each line orientation vector, $\mathbf{v_i}$, using a soft data assignment:

$$q_{it} = \left\| x(t) - (v_i . x(t)) v_i \right\|^2 \quad (4)$$

$$\tilde{q}_{it} = P(x(t) \mid v_i) = \frac{e^{-\gamma q_{it}}}{\sum_{i'} e^{-\gamma q_{i't}}} \quad (5)$$

Where q controls the boundary between the regions attributed to each line, and $\tilde{q}_{it}$ is the computed weightings of observation at time t for each line i.

d) Calculate the covariance matrix for the weighted observation assigned to each line. The covariance matrix expression and assignment weightings are combined as follows:

$$\Sigma_i = \frac{\sum_t \tilde{q}_{it} (x(t) - \mu)(x(t) - \mu)^{\mathrm{T}}}{\sum_t \tilde{q}_{it}} \quad (6)$$

Where μ is a vector of the mean values of the rows of x, which is typically zero for speech, and Σ_i is the covariance of weighted observation, related to line i.

e) Update the line orientation estimates to the principal eigen vectors of each covariance matrix: The eigen-vector decomposition of Σ_i is:

$$\Sigma_i = U_i \Lambda_i U_i^{-1} \quad (7)$$

Where the columns of U_i contain the eigen-vectors of Σ_i and Λ_i is the matrix containing the eigen values of it. The new estimation of the line orientation vector is the principal eigen vector of $\sum_i$ means: $v_i \leftarrow u_{\max}$ where $u_{\max}$ is the eigen vector corresponding to the largest eigen value $\lambda_{\max}$.

f) Update γ using the variances that are orthogonal to the direction of the lines: Select the second largest eigen-value from each diagonal matrix Λ_i, and update to the reciprocal of the largest eigen-value from this set, which λ_i below is the second eigen-value of $\sum_i$. Return to step 3 and repeat until v_i converges.

$$\gamma \leftarrow \frac{1}{\max(\lambda_i, ..., \lambda_M)} \tag{8}$$

g) After convergence, adjoin the line orientation estimations to form $\hat{A}$,

$$\hat{A} = [v_1 | ... | v_N] \tag{9}$$

Now after estimating the mixing matrix it is the turn to proceed separating the sources from the mixtures. The steps of this interval are as follows:

i. Knowing the estimated mixing matrix, $\hat{A}$, we try to find the minimum ℓ^2-norm solution of As = x. The solution is obtained using pseudo inverse.

ii. Then we choose a suitable decreasing sequence for σ, $[\sigma_1, ..., \sigma_l]$ which σ_i characterizes a trade off between the approximate accuracy and the smoothness of it, so that the smaller the σ, the more accurate the estimation of the sources and in an opposite situation the bigger the σ the smoother of the estimation will occur as it is mentioned in [9].

iii. In a loop which is consecutively done for the number of σ vector elements there are some functions to be done:

A. Maximize the function F_{σ_i} on the feasible set s using L iterations of the steepest descent algorithm, where F_{σ_i} is a function which approximates the l^0-norm of the sources in transformed domain [9].

B. Maximize $F_\sigma(s)$ subject to As = x, which $\sigma \to \infty$ is the minimum ℓ^2-norm solution of the equation A**s** = **x**. This procedure can be done in an iterated loop (internal loop) which choosing the number of iterations (L) here for obtaining better results is really crucial and will be explained later on. This internal loop is to perform steepest descent algorithm which is: $\boldsymbol{s} \leftarrow \boldsymbol{s} + \mu_i \nabla \boldsymbol{F}_\sigma(\boldsymbol{s})$ and μ is the learning rate of the algorithm. And the value of μ has to be changed downwards in each iteration. In other words for smaller σ_i, we have to apply smaller values of μ_i. This is because of the fact that for the smaller values of σ, the F_σ function is more fluctuating and therefore for maximizing it smaller values of step size parameter have to be chosen. After applying the steepest descent algorithm the value of the estimated sources are assigned to the vectors of v_i. Here the external loop iterates for the number of σ_i. These loops are implicitly doing the process of finding a solution for l^0-norm minimization problem which is ([9]):

$$\arg \min_{\hat{s}(\omega) \in \mathbb{R}^N} \left\| \hat{s}(\omega) \right\|_1 \; s.t. \; \hat{A}\hat{\mathbf{s}}(\omega) = \mathbf{X}(\omega) \tag{10}$$

iv. After finding the sources in the transformed domain an ISTFT should be done to obtain the time based sources, and the algorithm is now finished.

Now we need to explain some considerations required to run the algorithm accurately while involving speech sources as the test signals. One is that the value of μ_i (learning rate) should be reversely proportional to the number of iterations which are done for the steepest ascent algorithm that maximizes F_{σ_i}. This is because of the fact that for steepest ascent algorithm μ_i should be decreasing and for smaller σ's the function F_σ is more fluctuating. The other one is that for smaller σ's, F_σ contains a lot of local maxima. Therefore, it's really difficult to directly maximize this function for very small σ's. Hence, to have both the good approximation of the l^0-norm of the sources and for escaping from local maxima σ must be decreased gradually. Thus, we can enter the region near the maximizer of F_σ without being tangled in local maxima. The last issue to consider is that the SL0 algorithm starts by initializing the estimation of sparse solution with an optional value. However, the best initial value is min-ℓ^2-norm of the equation A**s** = **x**, which this value is obtained through pseudo–inversing the mixing matrix, A. In this case the value of **s** is, $\mathbf{s}=\mathbf{A}^T(\mathbf{AA}^T)^{-1}\mathbf{x}$ ([9]). After initializing the algorithm with a min-ℓ^2-norm solution, the next value for σ should be selected twice the absolute value of the calculated sources means, $\sigma_i = \max_i |s_i|$. The next value of σ is selected like this:

$$\sigma_i = \alpha \sigma_{i-1} \tag{11}$$

Where: $0.5 < \alpha < 1$

And the more close to 1 for α gives the better results.

Following the considerations mentioned above we are going to express the mathematical justifications behind the proposed algorithm and explain how it may work better than its antecedents (Soft/Hard LOST) or when it may have better performance while compared to some famous algorithms like DUET which is used in the same type of applications. Following this section the simulation results show the validity of these justifications.

For the under-determined case, which is usually restricted to sparse methods, a linear transformation is not possible since $\hat{A}\hat{\boldsymbol{s}}(\boldsymbol{t}) = \boldsymbol{x}(\boldsymbol{t})$ has more unknown parameters than known ones in **x(t)** and therefore is not invertible [2]. Furthermore, the Moore-Penrose pseudo-inverse, which corresponds to the minimum ℓ^2-norm solution, cannot be used as it is unable to remove the correlation between samples, and separate the mixtures. Consequently, some non-linear methods are needed to estimate the sources. These methods usually involve assigning the observed data **x(t)** to the columns of $\widehat{A}$, which characterizes each source. The most rudimentary method is to hard assign each data point to only one source based on some measure of proximity to columns of $\widehat{A}$ ([1]). A logical extension of this is the partial assignment of each data point to multiple sources. If the sources are sparse, it is desirable that the data point assignment results in one significantly non-zero coefficient ([1]), with the other coefficient values being close to zero. These kinds of assignments are shown by optimization problems, which are formulated by enforcing a constraint on the norm of the solution ([1]). Ideally, a sparse representation will have only one active coefficient, and the ℓ^0-norm measures the active coefficients in a vector. This solution produces a sparse representation. But, minimization of ℓ^0-norm cannot be computed in polynomial time and have some difficulties which are stated in [1]. Another alternative is using ℓ^1-norm minimization (sometimes referred to as Basis-Pursuit) method, which produces similarly sparse solutions. This method is a piecewise linear operation that partially assigns the energy of **x(t)** to the columns of $\widehat{A}$ that form a cone around **x(t)** in $\mathbb{R}^{M}$ space, with the remaining (N–M) columns assigned zero coefficients. When the number of sources active at any one time is less than or equal to M, more accurate source estimations are produced ([1]). This solution can be achieved by formulating the problem as a linear programming one. This is a fact that although the minimization of ℓ^0-norm is the main measure of sparsity, however when the dimensionality of the problem increases, for most of the linear under-determined system of equations minimization of ℓ^0-norm and ℓ^1-norm lead to a same result ([1]). In other words, because the ℓ^0-norm is the measure of sparsity, we can say that minimizing the ℓ^1-norm leads to the sparsest solution. In [9] it is proven that the method of *Smoothed l^0-norm minimization* can provide the same result as the previously mentioned ones, whereas it doesn't suffer the shortages which are inherently along with implementing the main measure of sparsity (means ℓ^0-norm). Moreover, in [10] it is proven that the separation performance is improved when one uses ℓ^q-basis pursuit with q<1 compared to the q =1 case.

The previously mentioned issues constructed the main idea behind the new algorithm which we called it SL0-LOST. Therefore, due to the formerly discussed issues we expect having better performance of separation when our algorithm is compared to Soft/Hard LOST. Furthermore, referring to [9] in case of speed, we expect that our algorithm would be much faster than its antecedents. Comparing SL0-LOST with DUET algorithm we can say that in DUET method, only the stage of clustering, which finds the cluster centers for labeling and forming the histogram, needs a loop of convergence. But in SL0-LOST method both stages have convergence loops. Especially, in the line orientation routine, finding the orientation of the lines in the scatter-plot, which leads to the estimation of the mixing matrix, is computationally a heavy routine [8]. Thus, it can be deduced that DUET be faster than our algorithm. But, in case of performance because DUET is confined to two mixtures it seems that when the number of sources increases, because the cross-terms increase drastically, separating the cluster centers become very difficult. Therefore, error of estimation in the mixing parameters increases and it affects the next stage of this algorithm (separation of sources). This problem is remaining unsolved for our algorithm, however structurally it is possible to have more than two mixtures and when the number of the sources increases the mixtures gradually become non-sparse. For non-sparse mixtures, solving the under-determined system of equations is indeed a linear embedding of an M-dimensional space into the higher N-dimensional one. Therefore, when the number of mixtures increases segregation of sources due to the line orientations become easier. That's why we expect that our algorithm would have better performance of separation when the number of microphones increases. At the next section we can see the simulation results and compare it with the former justifications mentioned.

V. Simulation Results

In this section we simulate our proposed algorithm and then compare it (in cases of performance and speed) with Soft/Hard LOST and DUET algorithms. Mixtures are generated synthetically. The algorithm tries to estimate the sources included in mixtures.

To evaluate the separation quality, signal to noise ratio (SNR) is used as an objective measure. Moreover, the CPU time is used as a measure of computation speed. Our simulations were performed in MATLAB 7.3 environment using an Intel P4 Core-Duo 1.86 GHz processor. The SNR values calculated from the algorithms are shown in the next figures.

As the SNR has ambiguity to the reader and never gives a real sense to the quality of separation and although defining perfect criteria for the correspondence between the objective parameter of SNR and the subjective criteria of quality is too hard ([11]), we have prepared a table (Table I) regarding the mixing sources which can fairly make this sense to the reader.

TABLE I
Setting Correspondence between SNR and Quality factors

Quality Factor	**Meaning**	**SNR(dB)**
1	Very Weak (useless)	$SNR < 1$
2	Weak (sometimes usable)	$1 < SNR < 4$
3	Good (usable)	$4 < SNR < 6$
4	Very good	$SNR > 6$

The results of comparison among 3 algorithms (DUET, LOST, SL0-LOST) in case of two mixtures are shown in figures [2], [3]. For the case of more than two mixtures where the DUET method cannot be used figure [4] (as a sample) shows the results.

In figures the mixtures and sources are abbreviated by "m" and "s" characters. Therefore 4s2m means the case of 4 sources and 2 mixtures. For this, the result of mentioned algorithms related to the including sources are shown in a column diagram. For example, in fig.2, for 3s2m, DUET can separate all the sources very Good. LOST extracts two of sources very Good and one of the sources Good and SL0-LOST extracts 1 source perfectly, one source Good and one source is not extracted. But for 6s2m LOST has the best performance and DUET has the worst (moderately).

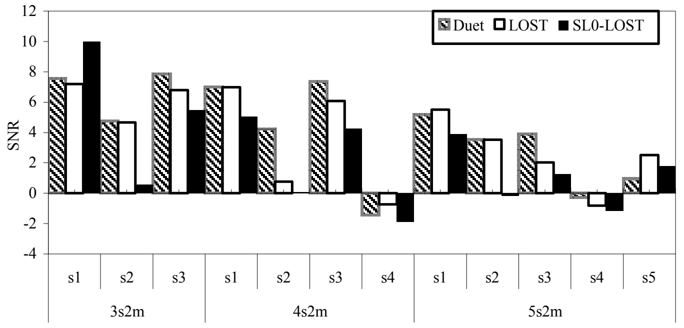

Figure 2. Comparison of Algorithms in case of two mixtures (s<6).

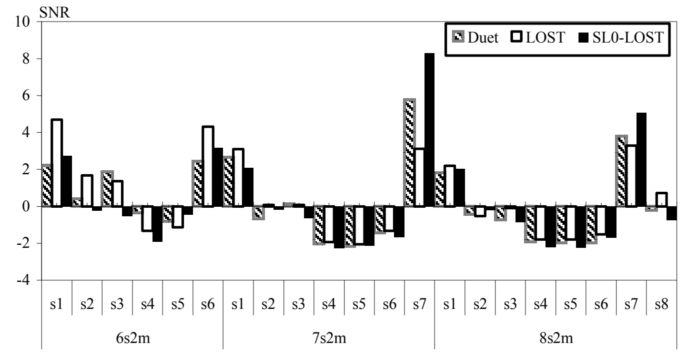

Figure 3. Comparison of Algorithms in case of two mixtures (s>6).

Figures [3], [4] show that by increasing the number of mixtures the performance of SL0-LOST is totally better than the others and this is the result we could expect in advanced. Totally, we can say the SL0-LOST method would extract some of the sources with very good performance and the other ones in nearly the same quality level of the other methods. This fact can be seen better when the number of mixtures increases.

There are some notifications about these tests. One is the negative amounts which can bee seen in figures [2-4]. That's because for a special source the algorithm has extracted another one. It means that turns of sources are missed.

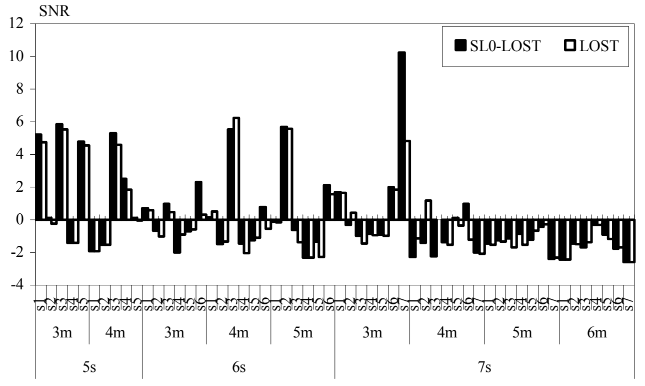

Figure 4. Results of objective test for more than two mixtures (s>5 , m>2).

Therefore we should only consider the positive values of SNRs which indicate the correct position of extractions. Also the more SNR for them imply the better quality of extraction. Second is that in cases where the number of mixtures is more than 2, DUET can't be used and comparisons are made between LOST and SL0-LOST. Therefore this is an advantage of this algorithm against DUET. Third is that rate for decreasing the quality of separation in SL0-LOST as seen in figures is slower than the others. Fourth is that DUET is only used for under-determined case while SL0-LOST and LOST are usable for Over/Even determined cases too ([7], [8]). Final advantage is that because SL0-LOST uses SL0 for separation stage it definitely carries all the advantages of this method comparing to the l^1-norm minimization of LOST. These advantages are described in [9] and practically can be seen when the number of mixtures increases.

The other remarkable issue is the CPU-time used for running these algorithms. Next table shows the exact time for running the algorithms and fig.5 denotes the relative time taken for their execution. We can see that the speed of source extraction from mixtures in SL0-LOST is between 3 to 10 times faster than LOST which is because of the inherent speed of SL0 method compared to L^1-norm minimization method ([9]).

TABLE II
SPEED OF SOURCE SEPARATION ALGORITHMS (PER SECOND)

mixing	**DUET**	**SOFT_LOST**	**SL0-LOST (New Algor.)**
3s2m	**10.7**	**298**	**28**
4s2m	**6**	**328**	**37**
5s2m	**6**	**389**	**100**
6s2m	**7**	**360**	**80**
7s2m	**9**	**408**	**89**
8s2m	**9**	**475**	**135**

And the next figure shows the computational speed comparison among the mentioned algorithms much better.

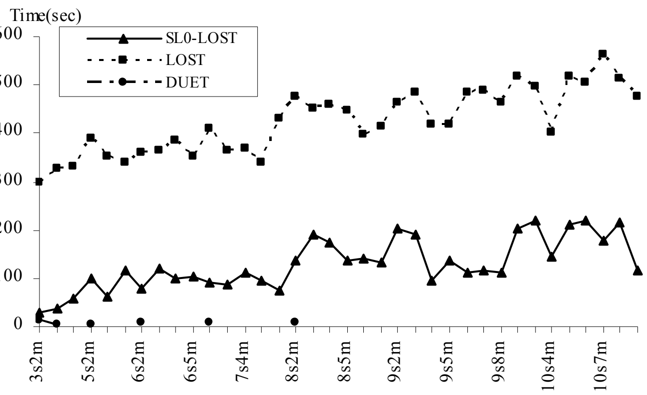

Figure 5. CPU-Time for all the algorithms mentioned in this article.

In DUET algorithm only the clustering section needs iteration of functions but in our method both stages use iterative functions to be executed in various loops to converge to suitable values and therefore is slower than DUET.

VI. Conclusion

Reviewing the previous section we can come to this point that our algorithm is totally more effective than the others when the number of mixtures increase. Furthermore, the rate of performance dropping is slower than the others when mixtures are more crowded. It is much faster than its predecessor but still slower than DUET which could be deduced due to the inherent features of these algorithms. Also it can be used for over/even determined cases but DUET can't be. Finally is the number of mixtures which is more than what can be used in DUET. All these issues are the remarkable advantages.

References

[1] P. O'Grady, B. A. Pearlmutter, S. T. Rickard, "Survey of Sparse and Non-Sparse Methods in Source Separation," *International Journal of Imaging Systems and Technology (IJIST)*, vol. 15, pp. 18–33, 2005.

[2] L. Vielva, and D. Erdoğmus, and J. C. Principe, "Underdetermined Blind Source Separation Using a Probabilistic Source Sparsity Model," in *2nd* International *Workshop on Independent Component Analysis Blind Signal Separation*, Jun. 2000.

[3] A. Jourjine, S. Rickard, and Ö. Yilmaz, "Blind Separation of Disjoint Orthogonal Signals: Demixing N Sources from 2 Mixtures," in *IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP2000)*, vol. 5, and pp. 2985-2988, September 4-8. 2006.

[4] F. Abrard, and Y. Deville, "'A time-frequency blind signal separation method applicable to under-determined mixtures of dependent sources," *Elsevier, Signal Processing, Feb. 2005, Pages 1389-1403.*

[5] T. Melia, and S. Rickard, C. Fearon, "Histogram-based blind source separation of more sources than sensors using DUET-ESPRIT technique," *in Proceedings of 13th European Signal Processing Conference (EUSIPCO '05),* Antalya, Turkey, September 2005.

[6] Ö. Yilmaz, and S. Rickard, "Blind Separation of Speech Mixtures via Time-Frequency Masking," in *IEEE Transaction on Signal Processing*, vol. 52, NO. 7, pp. 1830–1847, July 2004.

[7] P. O'Grady, B. A. Pearlmutter, "Hard-LOST: Modified k-Means for Oriented Lines" in *ISSC Conference,* Belfast, June 30 - July 2, 2004.

[8] P. O'Grady, B. A. Pearlmutter, "Soft-LOST: EM on a Mixture of Oriented Lines," in *ISSC Conference,* Belfast, June 30 - July 2, 2004.

[9] G. H. Mohimani, M. Babaie-zadeh, C. Jutten, "Fast Sparse Representation based on Smoothed ℓ^0 -Norm," in *ICA2007 Conference*, London, Sep. 2007.

[10] R.Saab, Ö. Yilmaz, M.J. McKeown , and R.Abugharbieh, "Underdetermined Anechoic Blind Source Separation via ℓ^q *-Basis Pursuit with q<1* ," in *IEEE Transaction on Signal Processing*, vol. 55, NO. 8, pp. 4004–4016, August 2007.

[11] E. Vincent, R. Gribonval, and C. Févotte, "Performance Measurement in Blind Audio Source Separation," in *IEEE Transaction on Audio, Speech, and Language Processing*, vol. 14, NO. 4, pp. 1462–1469, July 2006.