Hard-LOST: Modified k-Means for Oriented Lines

Paul D. O'Grady[†] **Barak A. Pearlmutter**[*]

Hamilton Institute
National University of Ireland Maynooth
Co. Kildare
Ireland

E-mail: [†]paul.ogrady@may.ie [*]barak@cs.may.ie

Abstract — **Robust clustering of data into linear subspaces is a common problem. Here we treat clustering into one-dimensional subspaces that cross the origin. This problem arises in blind source separation, where the subspaces correspond directly to columns of a mixing matrix. We present an algorithm that identifies these subspaces using a modified k-means procedure, where line orientations and distances from a line replace the cluster centres and distance from cluster centres of conventional k-means. This method, combined with a transformation into a sparse domain and an L_1-norm optimisation, constitutes a blind source separation algorithm for the under-determined case.**

I Introduction

We encounter a mixture of oriented lines in the context of linear source separation, in which a set of N sensor observations, $\mathbf{X} = (\mathbf{x}(1) \,|\cdots|\, \mathbf{x}(T))$, consist of a linear mixture of M source signals, $\mathbf{S} = (\mathbf{s}(1) \,|\, \cdots \,|\, \mathbf{s}(T))$, by way of an unknown linear mixing process characterised by the $N \times M$ mixing matrix $\mathbf{A}$. These measurements may be corrupted by additive noise ϵ,

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \epsilon(t) \tag{1}$$

When $N = M$ and $\epsilon \approx 0$ the underlying sources can be recovered if one can find an unmixing matrix $\mathbf{W}$,

$$\hat{\mathbf{s}}(t) = \mathbf{W}\mathbf{x}(t) \tag{2}$$

where $\hat{\mathbf{s}}(t)$ holds the estimated sources at time t and $\mathbf{W} = \mathbf{A}^{-1}$ up to permutation and scaling of the rows. Time-sampled vectors $\mathbf{x}(t)$ and $\mathbf{s}(t)$ are composed of samples $x_1(t), \ldots, x_N(t)$ and $s_1(t), \ldots, s_M(t)$, respectively. In the oriented lines separation case $\mathbf{S}$ can be thought of as individual sets of one-dimensional data and $\mathbf{X}$ can be thought of as a mixture representation of this data projected down to N-dimensions.

Linear mixing imposes a structure on the resultant mixtures which becomes apparent when the mixtures have a sparse representation. For sources of interest (voice, music) this can often be achieved by a transformation into a suitable basis such as such as the Fourier, Gabor or Wavelet basis. The existence of this structure can be explained as follows. Given a mixing matrix

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

it is evident that if only one source is active, say s_1, then the resultant mixtures would be

$$\mathbf{x}(t) = \begin{pmatrix} a_{11} \\ a_{21} \end{pmatrix} s_1(t)$$

therefore the points on the scatter plot of $x_1(t)$ versus $x_2(t)$ would lie on the line through the origin whose direction is given by the vector $(a_{11}, a_{21})^T$. When the sources are sparse, making it unusual for more than one source to be active at the same time, the scatter plot of coefficients would constitute a mixture of lines, with the lines broadened due to noise and the occasional simultaneous activity (see Figure 1.) The line orientations correspond to the columns of the mixing matrix $\mathbf{A}$, so if the lines can be estimated from the data then an estimate of the mixing matrix can be trivially constructed.

An algorithm for identification of radial line orientation and line separation is presented in Section II. The application of the algorithm to blind source separation of speech signals in both the even-determined and under-determined case, along with experimental results, are presented in Section III.

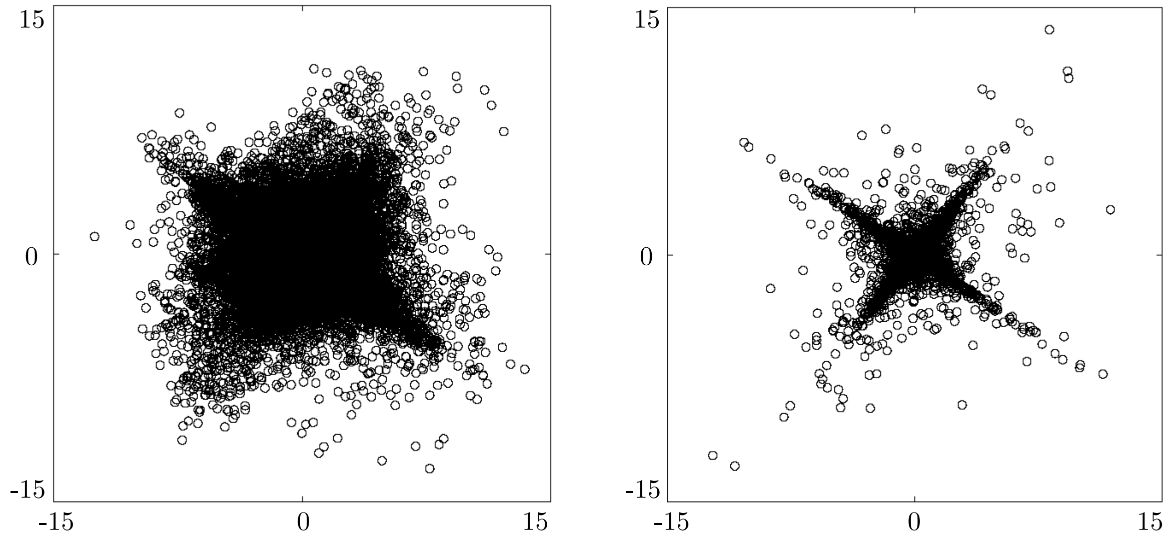

Fig. 1: Scatter plot of two linear mixtures of two zero-mean sources, in both the time domain **(left)** and the transform domain **(right)**. The "sparse" transform domain consists of the coefficients of 512-point windowed FFTs. The figures axises are measured in arbitrary units of transform domain mixture coefficients.

II Oriented Lines Separation

a) Determining Line Orientation

The orientation of a set of data points can be determined by a stochastic gradient algorithm [1, page 133] which finds the principal eigenvector of a data set. In order to identify various lines within a dataset containing a mixture of lines, it would be ideal to identify which data points come from each line, and run the stochastic gradient procedure separately on each segregated subset of data. Of course such identifications are not available, so instead we use a proxy by associating each data point with the estimated line most likely to have produced it (eq. 3). The stochastic gradient algorithms for the various lines are then run independently, with each point contributing to the estimation of the orientation of the line it is assigned to. Naturally these assignments change as the algorithm iterates, because the lines themselves are moving as they adjust to match the data. The stochastic gradient algorithm is initialised with line orientation vectors, $\mathbf{v}_i$, which are drawn random on the unit N-sphere by sampling an N-dimensional Gaussian and normalising the resulting vector. The estimated mixing matrix $\hat{\mathbf{A}}$ is formed by adjoining the estimated line orientations to form the columns of the matrix (eq. 5).

b) Data Point Separation

For the even-determined case where $N = M$ the estimated mixing matrix $\hat{\mathbf{A}}$ is square and the data points can be assigned to line orientations using (eq. 2). When $N < M$, the under-determined case, $\mathbf{A}$ is not invertible so the sources need to be estimated by some other means. One technique is so-called *hard assignment* of coefficients using a mask [2, 3]. Another is partial assignment, in which each coefficient can be decomposed into more than one source. This is generally done by minimisation of the L_1-norm, which can be accomplished by formulating the problem as a linear program.[1] This can be seen as a maximum likelihood reconstruction under the assumption that the coefficients are drawn from a distribution of the form $p(c) \propto \exp -|c|$, *i.e.* a Laplacian [4, 5].

c) Algorithm Summary

We present an algorithm called *Hard-LOST*, for *Line Orientation Separation Technique*. The prefix "hard" indicates that data points are assigned to lines using a *winner-takes-all* assignment. A discussion of hard and soft assignments is presented by Kearns et al. [7]. The algorithm is composed of a hard line orientation estimation subroutine which is called by the separation algorithm.

[1]The solution can be found efficiently using linear programming [6]. We introduce vectors $\mathbf{c}^+$ and $\mathbf{c}^-$, each with the same dimensionality as $\mathbf{c}$, and use the linear constraints

$$\begin{aligned} \mathbf{c}^+, \mathbf{c}^- &\geq 0 \\ \hat{\mathbf{A}}\mathbf{c}^+ - \hat{\mathbf{A}}\mathbf{c}^- &= \mathbf{d}_j \end{aligned}$$

The minimisation of $\|\hat{\mathbf{c}}\|_1 = \sum_{ij} |\hat{c}_{ij}|$ becomes the linear objective

$$\text{minimise} \sum_{ij} (\mathbf{c}^+_{ij} + \mathbf{c}^-_{ij})$$

After solving this system, the desired coefficients are

$$\hat{\mathbf{c}} = \mathbf{c}^+ - \mathbf{c}^-$$

When using complex data, as in the case of a FFT representation, we treat the real and imaginary parts separately, thus doubling the number of coefficients.

hard line orientation estimation

1. Randomly initialise the M line orientation vectors $\mathbf{v}_i$.
2. Assign each data point $\mathbf{d}_j$, where $\mathbf{d}_j = \mathbf{x}(j)$, to a line orientation vector using

$$z_{ij} = \|\mathbf{d}_j - (\mathbf{v}_i(t) \cdot \mathbf{d}_j)\, \mathbf{v}_i(t)\|^2 \qquad (3)$$

$$\hat{z}_{ij} = \begin{cases} 1 & \text{if } \forall i' \neq i,\ z_{i'j} < z_{ij} \\ 0 & \text{otherwise} \end{cases}$$

   $\hat{z}_{ij}$ is an indicator function for whether line i is the closest line to data point j.
3. Determine the line orientation vectors using the stochastic gradient algorithm with normalising constraint $\|\mathbf{v}_i\| = 1$.

$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) + \gamma(t)[y_i(t)\mathbf{d}_j - y_i^2(t)\mathbf{v}_i(t)] \qquad (4)$$
$$y_i(t) = \mathbf{v}_i(t)^T(z_{ij}\mathbf{d}_j)$$

   where $\mathbf{v}_i(t+1)$ is the current estimate of the direction of the data closest to line i and γ is the learning rate of the algorithm. Perform the calculation over all data points $\mathbf{d}_1 \ldots \mathbf{d}_T$. The line orientation estimates are taken from the final results $\mathbf{v}_i(T)$.
4. Adjoin the line orientations estimates to form the estimated mixing matrix.

$$\hat{\mathbf{A}} = [\mathbf{v}_1(T)| \cdots |\mathbf{v}_M(T)] \qquad (5)$$

This algorithm is a modified k-means procedure [8] where line orientation vectors and distances from a line replace cluster centres and distance from the cluster centre. Previously this has been accomplished using *fuzzy C-means* to cluster the data following a projection onto a unit hemisphere $\mathbf{d}_j \mapsto \text{sign}([1\,0 \cdots 0]\mathbf{d}_j)\,\mathbf{d}_j/\|\mathbf{d}_j\|$, or variants thereof [9, 10].

Hard-LOST line separation algorithm

1. Perform *hard line orientation estimation* to calculate $\hat{\mathbf{A}}$.
2. For the even-determined case data points are assigned to line orientations using (eq. 2). For the under-determined case calculate coefficients $\mathbf{c}_j$ using linear programming for each data point j such that

$$\text{minimise } \|\mathbf{c}_j\|_1 \text{ subject to } \hat{\mathbf{A}}\mathbf{c_j} = \mathbf{d_j}$$

   The resultant $\mathbf{c}_j$ coefficients, properly arranged constitute the estimated linear subspaces,

$$\hat{\mathbf{S}} = [\mathbf{c}_1| \cdots |\mathbf{c}_T]$$

3. The final result is a $M \times T$ matrix $\hat{\mathbf{S}}$ that contains the line orientation data sets in each row.

III Experimental Results

a) Hard-LOST in a BSS framework

The Hard-LOST algorithm is tested in a blind source separation framework where source estimates are analogous to decomposed linear subspaces and instantaneous linear mixtures are analogous to line mixtures in N-space. The Hard-LOST solution to BSS is

Hard-LOST for BSS

1. A $N \times T$ data matrix $\mathbf{X}(t)$ is composed of sensor observations of N instantaneous mixtures. The data is transformed into a sparse representation, $\mathbf{X}(t) \mapsto \mathbf{X}(\omega)$. M is the number of sources expected and γ controls the convergence rate
2. The Hard-LOST algorithm is performed on the data $\mathbf{X}(\omega)$. The algorithm estimates an unmixing matrix that will allow individual sources to be estimated from the mixtures.
3. The resultant $M \times T$ matrix $\hat{\mathbf{S}}(\omega)$ contains in its rows the M estimated sources $\hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_M$. These estimates are then transformed back into the time domain, $\hat{\mathbf{S}}(\omega) \mapsto \hat{\mathbf{S}}(t)$.

b) Error Measurement

The Signal-to-Noise Ratios of the estimated sources $\hat{\mathbf{s}}_i$ (in dB), $\text{SNR}_i = 20\log_{10}\|\mathbf{s}_i\|/\|\hat{\mathbf{s}}_i - \mathbf{s}_i\|$, are used to measure the performance of the above algorithm.

c) Experimental Method

Speech signals (see Figure 2 and Appendix A) were transformed using a 512-point windowed FFT and the real coefficients were used to create a scatter plot. Figure 1 illustrates that the transformation results in sparse data, with the source line orientations becoming more defined. A simple example of a scatter plot for the under-determined case is presented in Figure 3. The experiments were coded for Matlab 6.5.0 and run on a 3.06 GHz Intel Pentium-4 based computer with 768MB of RAM. Experiments for the under-determined case typically took 30 minutes while the tests for the even-determined case ran for less than 3 minutes. The effectiveness of separation in the under-determined case where L_1-norm minimisation was used, is evaluated by presenting the algorithm with the original $\mathbf{A}$ matrix used in the mixing process. In these experiments the line orientation estimation phase is skipped and the L_1-norm minimisation phase is tested separately. In general the better defined the line orientations in the scatter plot, the more accurate the source estimates. Experiments were performed for a range of different

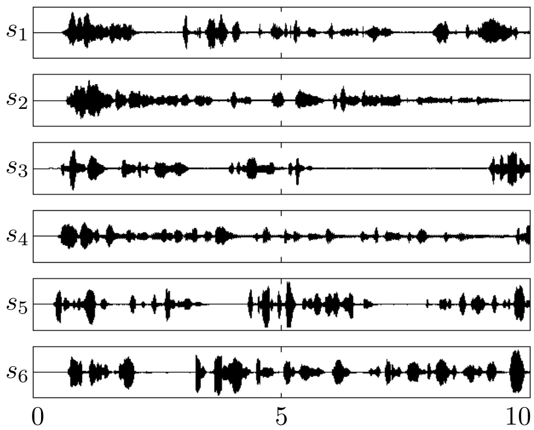

Fig. 2: Ten-second clips of six acoustic sources. Sound wave pressure is plotted against time and units are measured in seconds (see Appendix A.).

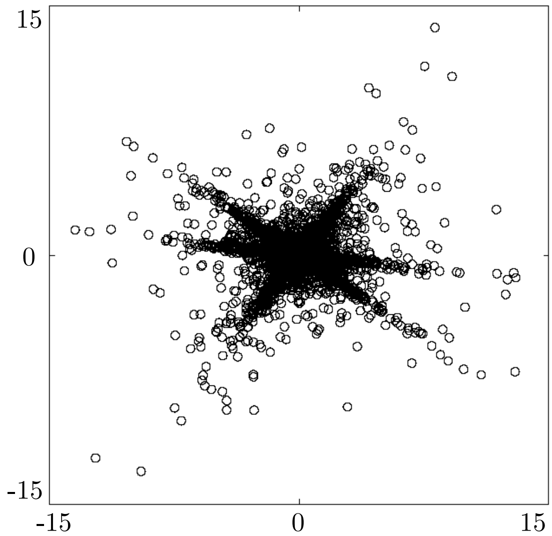

Fig. 3: 512-point FFT Scatter plot for two linear mixtures of three sources.

values of N and M, and the parameter γ was varied on an ad-hoc basis.

d) Results

Results are presented for a total of 18 experiments. Data on the number of mixtures, sources used, and the value of the learning parameter γ are contained in the tables of results. Results in tables 1 and 2 demonstrate the effectiveness of the algorithm for the even-determined case. Experiments for testing line separation using L_1-norm minimisation were performed and their results are presented in table 3. These experiments evaluate the effectiveness of the separation phase of the Hard-LOST algorithm in the under-determined case, and provide a benchmark for the subsequent experiments.

Results for experiments that test both line orientation estimation and line separation in the under-determined case are presented in tables 4 and 5, with some additional results in table 6. The Hard-LOST algorithm was tested for robustness to noise. Gaussian noise of various intensities was added to the signals of the experiments in table 7, where the noise introduced to each signal is measured in terms of SNR values. These results, when contrasted with those previously presented, indicate the algorithm's robustness to noise.

The experimental results demonstrate that the Hard-LOST algorithm is an effective technique for BSS in both the even-determined and under-determined case, even in the presence of noise.

IV Conclusion

The results presented demonstrate that a modified k-Means procedure is able to identify scatter plot line orientations, thus determining the mixing matrix of a set of linear mixtures. It has been shown that once the mixing matrix is found, sources can then be separated by minimising the L_1-norm between the data point being considered and the line orientations represented by the columns of the mixing matrix. The Hard-LOST algorithm provides a good solution to blind source separation of instantaneous mixtures even when there are fewer sensors than sources. The algorithm is scalable to a large number of sensors and sources because L_1-norm minimisation can be reduced to a linear programming problem. The experiments presented are concerned with the specific problem of blind source separation of speech signals, however the results can be applied to any situation involving a mixture of oriented lines.

We plan to extend the line orientation estimation phase using a soft data point assignment based on a modified EM algorithm, and to replace the stochastic gradient calculation of eq. 4 with a batch mode calculation based on covariance matrix eigenvector decomposition.

Acknowledgements

Supported by Higher Education Authority of Ireland (An tÚdarás Um Ard-Oideachas) and Science Foundation Ireland grant 00/PI.1/C067.

References

[1] Juha Karhunen Aapo Hyvärinen and Erkki Oja. *Independent Component Analysis.* John Wiley & Sons, 2001.

[2] S. T. Rickard and F. Dietrich. DOA estimation of many W-disjoint orthogonal sources from two mixtures using DUET. In *Proceedings of the 10th IEEE Workshop on Statistical Signal and Array Processing (SSAP2000)*, pages 311–314, Pocono Manor, PA, August 2000.

[3] Sam T. Roweis. One microphone source sep-

aration. In *Advances in Neural Information Processing Systems 13*, pages 793–799. MIT Press, 2001.

[4] T.-W. Lee, M. S. Lewicki, M. Girolami, and T. J. Sejnowski. Blind source separation of more sources than mixtures using overcomplete representations. *IEEE Signal Processing Letters*, 6(4):87–90, 1999.

[5] Michael Zibulevsky and Barak A. Pearlmutter. Blind source separation by sparse decomposition in a signal dictionary. *Neural Computation*, 13(4):863–882, April 2001.

[6] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1999.

[7] M. Kearns, Y. Mansour, and A. Y. Ng. An information-theoretic analysis of hard and soft assignment methods for clustering. *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 282–293, 1997.

[8] J. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297, Berkeley, 1967. University of California Press.

[9] P. Bofill and M. Zibulevsky. Underdetermined blind source separation using sparse representations. *Signal Processing*, 81(11):2353–2362, 2001.

[10] Michael Zibulevsky, Pavel Kisilev, Yehoshua Y. Zeevi, and Barak A. Pearlmutter. Blind source separation via multinode sparse representation. In *Advances in Neural Information Processing Systems 14*, pages 1049–1056. MIT Press, 2002.

[11] Elise Paschen, Charles Osgood, and Rebekah Presson Mosby, editors. *Poetry Speaks: Hear Great Poets Read Their Work from Tennyson to Plath.* Sourcebooks Incorporated, 2001. ISBN 1570717206.

Table 1: Two Mixtures and Two Sources

Mixtures	Sources	γ	SNR (dB)	
2	s_1, s_2	0.0005	31.35	39.88
2	s_3, s_4	0.0005	38.47	45.89
2	s_5, s_6	0.0005	21.94	22.54

Table 2: Five Mixtures and Five Sources

Mixtures	Sources	γ	SNR (dB)	
5	s_1, s_2	0.005	21.35	32.78
	s_3, s_4		20.46	28.59
	s_5		25.80	
5	s_1, s_2	0.005	21.65	32.77
	s_3, s_4		20.65	28.58
	s_5		25.81	
5	s_1, s_2	0.005	25.79	32.64
	s_3, s_4		24.75	28.52
	s_5		25.93	

Table 3: L_1-Norm and True Mixing Matrix

Mixtures	Sources	SNR (dB)		
2	s_1, s_2, s_3	10.41	15.64	7.75
5	s_1, s_2, s_3	20.85	20.62	19.1
	s_4, s_5, s_6	17.08	21.93	48.96

Table 4: Two Mixtures and Three Sources

Mixtures	Sources	γ	SNR (dB)	
2	s_1, s_2	0.0005	7.05	12.9
	s_3		3.44	
2	s_1, s_2	0.0005	10.41	15.59
	s_3		7.68	

Table 5: Five Mixtures and Six Sources

Mixtures	Sources	γ	SNR (dB)	
5	s_1, s_2	0.005	20.02	19.90
	s_3, s_4		17.76	15.92
	s_5, s_6		18.12	27.12
5	s_1, s_2	0.005	20.28	20.14
	s_3, s_4		18.22	16.57
	s_5, s_6		18.77	28.97

Table 6: Four Mixtures and Five & Six Sources

Mixtures	Sources	γ	SNR (dB)	
4	s_1, s_2	0.005	17.22	16.37
	s_3, s_4		16.40	13.50
	s_5		25.35	
4	s_1, s_2	0.005	8.64	6.26
	s_4, s_5		10.35	7.02
	s_5, s_6		14.62	10.33

Table 7: Additive Gaussian Noise

Mixtures	Sources	noise	SNR (dB)	
2	s_1, s_2	5 dB	24.23	24.82
4	s_1, s_2	10 dB	11.95	11.77
	s_3, s_4		12.23	8.26
	s_5		20.82	
5	s_1, s_2	15 dB	19.85	28.77
	s_3, s_4		21.00	27.68
	s_5		25.19	
5	s_1, s_2	15 dB	16.73	16.87
	s_3, s_4		15.29	13.50
	s_5, s_6		16.79	29.46

A Source Signals

The source signals were taken from a commercial audio CD of poems read by their authors [11]. Such data is recorded as raw 44.1 kHz 16-bit stereo waveforms. Prior to further processing ten-second clips were extracted, the two signal channels were averaged, and the data was down-sampled to 8 kHz. The scale of the audio data is arbitrary, leading to the arbitrary units on auditory waveform samples throughout the manuscript.

- s_1 *Coole Park and Ballylee*, by William Butler Yeats.
- s_2 *The Lake Isle of Innisfree*, by William Butler Yeats.
- s_3 *Among Those Killed in the Dawn Raid Was a Man Aged a Hundred*, by Dylan Thomas.
- s_4 *Fern Hill*, by Dylan Thomas.
- s_5 *Ave Maria*, by Frank O'Hara.
- s_6 *Lana Turner Has Collapsed*, by Frank O'Hara.