Robust ICA for Super-Gaussian Sources

Frank C. Meinecke[1], Stefan Harmeling[1], and Klaus-Robert Müller[1,2]

[1] Fraunhofer FIRST, IDA group, Kekuléstr. 7, 12489 Berlin, Germany
{meinecke,harmeli,klaus}@first.fhg.de
[2] University of Potsdam, Department of Computer Science, August-Bebel-Strasse 89, 14482 Potsdam, Germany

Abstract. Most ICA algorithms are sensitive to outliers. Instead of robustifying existing algorithms by outlier rejection techniques, we show how a simple outlier index can be used directly to solve the ICA problem for super-Gaussian source signals. This ICA method is outlier-robust by construction and can be used for standard ICA as well as for over-complete ICA (i.e. more source signals than observed signals (mixtures)).

1 Introduction

ICA models multi-variate time-series $x_n(t)$ with $n = 1 \dots N$ as a linear combination of statistically independent source signals $s_m(t)$ with $m = 1 \dots M$:

$$x_n(t) = \sum_m A_{nm} s_m(t). \tag{1}$$

The task of an ICA algorithm is to estimate the *mixing matrix* A given only the observations $x(t)$. Typically, it is assumed that $M \leq N$ and that the columns of A are linearly independent. In this case, Eq. (1) is invertible and the source signals $s(t)$ can be recovered[3].

In the over-complete[4] case, where the number of sources exceeds the number of mixtures (i.e. $M > N$), it is often (if the sources are supergaussian or sparse) still possible to identify the mixing matrix A. However, in general the source signals cannot be recovered, since the model Eq. (1) is not invertible. For very sparse signals (or signals that can be represented sparsely, [1–3]) the underdetermined blind source separation problem is solvable, because each data point can be uniquely assigned to one source (at least approximately).

There exists many of algorihms that can solve the task of estimating the mixing matrix A. Most of them make use of statistical properties of the projections (i.e. kurtosis, negentropy, time lagged covariance matrices...). However, most existing ICA algorithms are highly sensitive to outliers (especially algorithms that employ higher-order statistics).

[3] Of course, the source signals can be recovered only up to scaling and permutation, since a scalar factor can be exchanged between each source and the corresponding column of A without changing $x(t)$. The numbering of the sources (and the columns of A) has no physical interpretation and is nothing but a notational device.

[4] also called under-determined

Recently, Harmeling et al. [4] proposed an outlier detection method based on indices that sort data from very typical points (inliers) to very untypical points (outliers). A simple strategy to robustify existing algorithms is to use these indices for outlier rejection. This is indeed possible, as shown in section 3.2. Moreover, we show that an appropriately defined outlier index can be used *directly* to solve the ICA problem for super-Gaussian source signals. The idea is to look for 'inliers' rather than outliers and use them as estimators for the ICA directions (i.e. columns of the mixing matrix A). Figure 1 shows a scatter

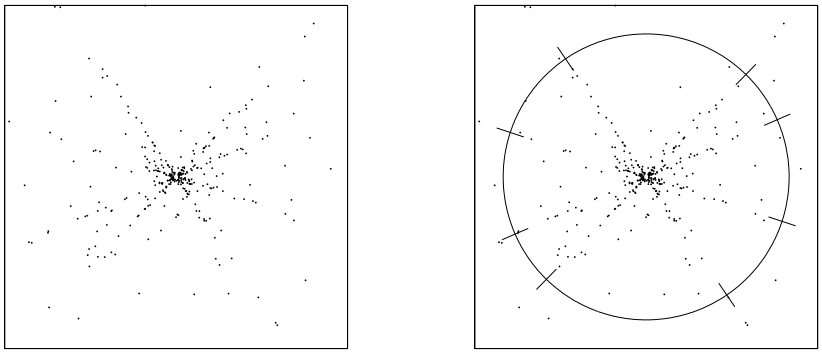

Fig. 1. The left panel shows a scatterplot of a two dimensional mixture of four super-Gaussian source signals. The right panel shows additionally the directions of the points of highest density on the unit circle. Those directions correspond to the columns of A.

plot of a two-dimensional mixture of four super-gaussian source signals (left and right panel). The colums of the mixing matrix are clearly visible as directions in the data space with higher density (right panel). To find these directions, we define a variation of the outlier index γ ([4]) that sorts the data points from very dense ('inlier') to very sparse ('outlier'). The inlier points are estimators for the columns of A. Since the scaling of the columns is arbitrary, also γ must ignore the scaling. This implies two requirements for the index:

- γ must be invariant under rescaling, i.e. $\gamma(\alpha v) = \gamma(v)$ for $\alpha > 0$.
- γ must be invariant under inversion, i.e. $\gamma(-v) = \gamma(v)$.

In other words, 'dense' or 'sparse' should be defined with respect to a distance measure between the *directions* of the data points (i.e. angle distance).

2 The Algorithm

We now describe the Inlier-Based ICA algorithm (abbr. IBICA). Note that some of the presented ideas appeared earlier in other geometrical algorithms (see [2, 5]). The main difference of IBICA is its usage of the inlier index which makes it particularly robust and allows it to be used even in high dimensions.

Step 1: project the data on the unit sphere.

Project all data points $x(1), \ldots, x(T)$ onto the unit sphere by normalizing to length one,

$$z(t) = \frac{x(t)}{\sqrt{x(t)^\top x(t)}} = \frac{x(t)}{|x(t)|}.$$

This step ensures the needed scaling invariance; the distances between the points $z(t)$ on the unit sphere do not depend on the scaling of the original points $x(t)$ but only on the directions. The ICA directions are now given by the dense regions on the sphere. Note that some fraction of the points at the disc around zero have to be removed, because in noisy settings these points do not contain much information about the correct signal directions (and we avoid division by zero for points exactly from the origin).

Step 2: calculate γ for an inversion invariant distance.

The natural distance measure (angle distance) between two normalized points a and b is the geodesic distance on the unit sphere, but we will use a distance measure based on the Euclidean distance since it is easier to calculate and yields similar results[5],

$$d(a, b) = \min(|a - b|, |a + b|).$$

This distance is invariant under the inversion operation (which maps a vector v onto $-v$). This is the natural distance measure to use for our problem, since we are not interested in the orientation of a vector.

Let now $\mathrm{nn}_1(z), \ldots, \mathrm{nn}_k(z)$ be the k nearest neighbors of z according to the distance d. We call the average distance of z to its k nearest neighbors the γ index of z, i.e.

$$\gamma(z) = \frac{1}{k} \sum_{j=1}^{k} d(z, \mathrm{nn}_j(z)).$$

Intuitively speaking, $\gamma(z)$ is large if z lies in a sparse region (z is probably an outlier), and $\gamma(z)$ is small if z lies in a dense region. The data points with the smallest z are good candidates for the directions of the signals, i.e. for the columns of A. We call these points inliers.

Step 3: pick the signal directions among the points with small γ

In order to obtain an estimate for the mixing matrix A, the first idea that comes to mind is to pick the M directions with the smallest values of γ and stack them

[5] For two points a and b on the unit sphere ($|a| = |b| = 1$) the geodesic distance is the angle between those vectors, i.e. $\arccos(a^\top b)$. However, for small angles this distance is proportional to the Euclidean distance, $|a-b| = \sqrt{(a-b)^\top (a-b)}$, and in general the relationship is monotonic, i.e. $\arccos(a^\top b) < \arccos(a^\top c) \Leftrightarrow |a - b| < |a - c|$ for another unit vector c.

together. The problem with this approach is that those M columns of A might originate all from the same direction, which by chance happened to be denser than the other directions. To be able to deal with such situations, we need a heuristic that avoids to pick a direction that is similar to a direction that has already been chosen.

Step 3a: deflational In the standard ICA setting (i.e. square mixing matrix), this is no problem, since it is possible to find the columns of A one after another in a deflation style: After whitening of the data, the γ values are calculated. The data point with the smallest γ is the first column of the estimated mixing matrix $\hat{A}$. The data set is projected onto the orthogonal subspace and the γ values are re-calculated. The next column of $\hat{A}$ is again given by the smallest gamma and so on. This ensures, that each column of A captures a different source signal since the search is always restricted to a subspace that is orthogonal to the one spanned by the directions found before.

Step 3b: symmetric If there are more source signals than mixtures, or if one would like to avoid the whitening step, the deflation procedure is not applicable.

The point density on the sphere (and therefore the distribution of γ values) peaks around the directions of interest. Our task is to find exactly one representative for each of the peaks with (locally) minimal γ. Therefore, after chosing a direction (i.e. a data point) with (globally) minimal γ, the data points forming the corresponding peak should be removed. These are all the data points that can be reached from the γ-minimum along the k-nearest-neighbor graph in a monotonically increasing sequence of γ. This idea is implemented in the following algorithm:

GREEDY PEAK SEARCH
- start with an empty matrix A
- put all points in the pool
- WHILE the pool is not empty
  - pick the point p from the pool with the smallest γ
  - store p as a new column of A
  - color p
  - WHILE there exist colored points in the pool
    - pick a colored point q from the pool
    - remove q from the pool
    - color the k nearest neighbors of q that have a larger γ than q and that are still in the pool
  - END
- END

Figure 2 shows the γ-landscape over the angle in the region $[-\frac{\pi}{2}, \frac{\pi}{2}]$ for an example of a two-dimensional mixture of four super-Gaussian sources for 10

and 50 nearest neighbors. Both figures show four pronounced peaks, but in the left panel the landscape is less smooth and and it has additional local minima. Using the heuristic with $k = 10$, more than four directions (shown as circles) are chosen. The choice of k influences, how many columns the estimated mixing matrix A has. Taking into account more neighbors (see the middle panel with $k = 50$), the γ-landscape is smoother and less components are chosen.

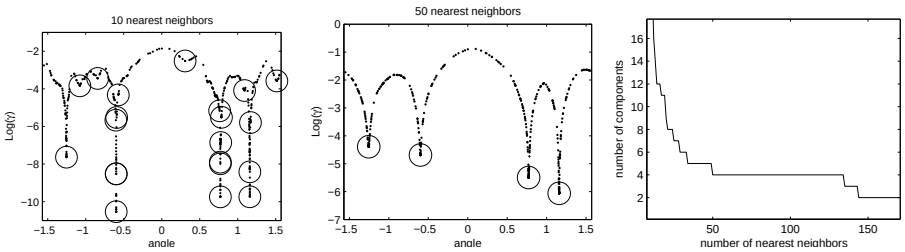

Fig. 2. The γ-Landscape of a two dimensional mixture of four source signals using 10 nearest neighbors (left) or 50 nearest neighbors (middle). In the first case, 21 directions (circles, see text for explanation) have been found, in the second only 4. In the first case the algorithm will therefore return a 2×21 mixing matrix, in the second the (correct) 2×4 Matrix. If the number of sources is not known in advance, one could try several k and look for a plateau (right).

If the number of components M in the mixture is known in advance, we can search for the smallest k that leads to M directions. This can be done very efficiently since the distance matrix has to be calculated and sorted only once. The choice of k influences only the calculation of γ. On the other hand, if the number of components is not known, the algorithm can be repeated efficiently (see previous paragraph) for several choices of k. By looking for a plateau in the number of chosen directions (i.e. by looking for a longer range of values of k that yield the same number of sources) a meaningful k can be found (see Fig. 2).

2.1 Speeding up IBICA

Since the computational costs of calculating the distance matrix grows quadratically with the number of data points, it is appropriate to divide big data sets into smaller subsets, calculate the γ on each of them, and keep only the best data points (i.e. those with the smallest γ) from each subset. Depending on the size of the data set and its subsets, the speed of IBICA can thus be significantly improved. Another side-effect is that this procedure makes IBICA more noise robust. When it comes to the final ICA step, the worst outliers are already removed. This reduces particularly the error, that is made by the whitening in the deflation mode of the algorithm. In the following experiments, we divide the data sets such that we have to deal with distance matrices of size of at most 1000×1000.

3 Experiments

3.1 Performance measures

To compare our algorithm with other standard ICA algorithms, we will use the following performance measure: Assume, that both the mixing matrix A and its estimator $\hat{A}$ are column normalized (i.e. the norm of the columns of these matrices is one). We then define:

$$pm(A, \hat{A}) = 1 - \left(\frac{1}{2M} \sum_{i=1}^{M} \max_j |A^\top \hat{A}|_{ij} + \frac{1}{2M} \sum_{j=1}^{M} \max_i |A^\top \hat{A}|_{ij} \right)$$

This performance measure is symmetrical ($pm(A, \hat{A}) = pm(\hat{A}, A)$), smaller or equal to 1 and zero only if $\hat{A} = AP$ with P being a permutation matrix (i.e. perfect solution).

3.2 Robustness against outliers

In the following experiments, we produc super-Gaussian source signals by taking gaussian noise to the power of three. The data sets contain 7000 data points each. We compare our algorithm (IBICA) with JADE [6] and FastICA [7].

First, we test the robustness against outliers. We mix two-dimensional super-Gaussian source signals with randomly chosen mixing matrices. Without outliers, the performances of IBICA, JADE and FastICA are all excellent (performance index ≈ 0.01). To test for outlier-robustness, we replace 50 data points with outliers, i.e. uniformly distributed data points within a disc of radius 500 around the origin (the norm of the original data points is roughly within the range from zero to 100).

As expected, IBICA still works fine. In fact, typically it does not even change its solution, because it simply ignores the outliers. JADE and FastICA however, produce arbitrary results because outliers can create directions of high kurtosis, which are attractive for algorithms that use higher order statistics.

3.3 Robustness against super-Gaussian noise

In the next experiment we add noise to the mixtures according to

$$x(t) = As(t) + \sigma\eta(t)$$

with $\eta(t)$ being a N-dimensional noise source of unit variance. We track the evolution of the perfomance index as a function of the noise level σ for kurtotic noise (we used multi-dimensional Gaussian noise, where we change the absolute value to the power of 5) in two dimensions and in 10 dimensions. Figure 3 shows that JADE and FastICA start to fail at a certain noise level, whereas IBICA continues to produce good ICA solutions. In the low dimensional case this difference is more pronounced than in higher dimensions, but even in 10 dimensions

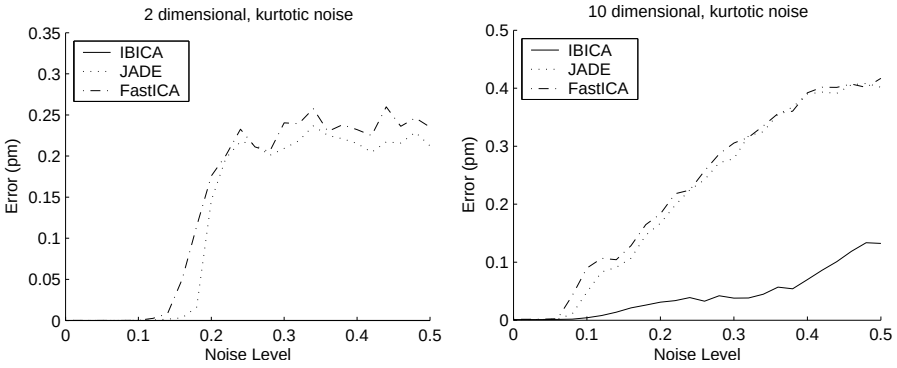

Fig. 3. Performance-index vs. noise level for kurtotic noise in two-dimensional (left) and 10-dimensional mixtures (shown is the median of 50 runs).

IBICA is still clearly superior. Note, that we have chosen the median over 50 runs because the separation performance of the algorithms depend strongly on the actual realization of the noise. However, the signals presented to the different algorithms are of course always the same.

3.4 Overcomplete ICA

As a last experiment, we will now test the ability of IBICA to solve overcomplete ICA problems. We will use the same data sets as before (super-Gaussian signals, 7000 data points). In order to reconstruct the source signals in an overcomplete setting, it is not enough to estimate the mixing matrix. In principle, a source signal reconstruction is only possible if the signals are sparse. There exists a number of techniques than can sparsify certain signals (see, e.g. [1, 8, 3]). However, here we simply assume, that the data can be sparsified by a suitable preprocessing step and focus only on the estimation of A.

We start with a two-dimensional mixture of four source signals (again 7000 data points) (see Figs. 1 and 2). The error is typically at $pm \approx 10^{-5}$, which is a perfect reconstruction of the mixing matrix.

The next example is a five-dimensional mixture containing 20 signals. Here, the error is at $pm \approx 0.01$. The largest angle deviation between one of the 20 source directions and their respective estimators is only about 1.5 degree.

4 Conclusion

Obtaining robust meaningful decompositions is essential when applying blind source separation techniques to data from the real world (see e.g. [9]). In most applications the data is strongly contaminated with measurement noise and outliers where unusual events not belonging to the probability distribution of interest or non-standard noise are measured. Such outlier events pose a severe problem to

most existing ICA algorithms, especially the ones that optimize kurtosis-based indices. Our contribution – besides pointing out this fundamental issue – is to use 'inlier' data points only, for performing the decomposition. As this novel framework for ICA does not depend on the dimensionality of the problem, it can be readily used also in overcomplete/underdetermined scenarios. Simulations underline these insights.

Future research will continue the quest for more robust blind source separation algorithms that can have a wider practical applicability.

Acknowledgement

The authors would like to thank Andreas Ziehe, Motoaki Kawanabe and Christin Schäfer for valuable discussions. This research has been partly supported by the PASCAL network of excellence (IST-2002-506778).

References

1. Zibulevsky, M., Pearlmutter, B.A.: Blind source separation by sparse decomposition in a signal dictionary. Neural Computation **13** (2001) 863–882
2. Bofill, P., Zibulevsky, M.: Underdetermined blind source separation using sparse representations. Signal Processing **81** (2001) 2353–2362
3. Lee, T.W., Lewicki, M., Girolami, M., Sejnowski, T.: Blind source separation of more sources than mixtures using overcomplete representations. IEEE Signal Process. Lett. **6** (1999) 78–90
4. Harmeling, S., Dornhege, G., Tax, D., Meinecke, F., Müller, K.R.: From outliers to prototypes: ordering data. Technical report (2004)
5. Puntonet, C.G., Prieto, A., Jutten, C., Rodriguez-Alvarez, M., Ortega, J.: Separation of sources: A geometry-based procedure for reconstruction of n-valued signals. Signal Processing **46** (1995) 267–284
6. Cardoso, J.F., Souloumiac, A.: Blind beamforming for non Gaussian signals. IEEE Proceedings-F **140** (1993) 362–370
7. Hyvärinen, A., Oja, E.: A fast fixed-point algorithm for independent component analysis. Neural Computation **9** (1997) 1483–1492
8. Chen, S., Donoho, D., Saunders, M.: Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing **20** (1998) 33–61
9. Meinecke, F., Ziehe, A., Kawanabe, M., Müller, K.R.: A resampling approach to estimate the stability of one-dimensional or multidimensional independent components. IEEE Transactions on Biomedical Engineering **49** (2002) 1514–1525