Scaling laws and local minima in Hebbian ICA

Magnus Rattray and Gleb Basalyga
Department of Computer Science, University of Manchester, Manchester M13 9PL, UK.
magnus@cs.man.ac.uk, basalygg@cs.man.ac.uk

Abstract

We study the dynamics of a Hebbian ICA algorithm extracting a single non-Gaussian component from a high-dimensional Gaussian background. For both on-line and batch learning we find that a surprisingly large number of examples are required to avoid trapping in a sub-optimal state close to the initial conditions. To extract a skewed signal at least \(O(N^2)\) examples are required for \(N\)-dimensional data and \(O(N^3)\) examples are required to extract a symmetrical signal with non-zero kurtosis.

1 Introduction

Independent component analysis (ICA) is a statistical modelling technique which has attracted a significant amount of research interest in recent years (for a review, see Hyvärinen, 1999). The goal of ICA is to find a representation of data in terms of a combination of statistically independent variables. A number of neural learning algorithms have been applied to this problem, as detailed in the aforementioned review.

Theoretical studies of ICA algorithms have mainly focussed on asymptotic stability and efficiency, using the established results of stochastic approximation theory. However, in practice the transient stages of learning will often be more significant in determining the success of an algorithm. In this paper a Hebbian ICA algorithm is analysed in both on-line and batch mode, highlighting the critical importance of the transient dynamics. We find that a surprisingly large number of training examples are required in order to avoid trapping in a sub-optimal state close to the initial conditions. To detect a skewed signal at least \(O(N^2)\) examples are required for \(N\)-dimensional data, while \(O(N^3)\) examples are required for a symmetric signal with non-zero kurtosis. In addition, for on-line learning we show that the maximal initial learning rate which allows successful learning is unusually low, being \(O(N^{-\frac{3}{2}})\) for a skewed signal and \(O(N^{-2})\) for a symmetric signal.

In order to obtain a tractable model, we consider the limit of high-dimensional data and study an idealised data set in which a single non-Gaussian source is mixed into a large number of Gaussian sources. Recently, one of us considered a more general model in which an arbitrary, but relatively small, number of non-Gaussian sources were mixed into a high-dimensional Gaussian background (Rattray, 2002). In that work a solution to the dynamics of the on-line algorithm was obtained in closed form for \(O(N)\) learning iterations and a simple solution to the asymptotic dynamics under the optimal learning rate decay was obtained. However, it was noted there that modelling the dynamics on an \(O(N)\) timescale is not always appropriate, because the algorithm typically requires much longer in order to
escape from a class of metastable states close to the initial conditions. In order to elucidate this effect in greater detail we focus here on the simplest case of a single non-Gaussian source and we will limit our analysis to the dynamics close to the initial conditions.

In recent years a number of on-line learning algorithms, including back-propagation and Sanger’s PCA algorithm, have been studied using techniques from statistical mechanics (see, for example, Biehl (1994); Biehl and Schwarze (1995); Saad and Solla (1995) and contributions in Saad (1998)). These analyses exploited the “self-averaging” property of certain macroscopic variables in order to obtain ordinary differential equations describing the deterministic evolution of these quantities over time in the large \( N \) limit. In the present case the appropriate macroscopic quantity does not self-average and fluctuations have to be considered even in the limit. In this case it is more natural to model the on-line learning dynamics as a diffusion process (see, for example, Gardiner, 1985).

2 Data Model

In order to apply the Hebbian ICA algorithm we must first sphere the data, i.e. linearly transform the data so that it has zero mean and an identity covariance matrix. This can be achieved by standard transformations in a batch setting or for on-line learning an adaptive sphering algorithm, such as the one introduced by Cardoso and Laheld (1996), could be used. To simplify the analysis it is assumed here that the data has already been sphered.

Without loss of generality it can also be assumed that the sources each have unit variance. Each data point \( x \) is generated from a noiseless linear mixture of sources which are decomposed into a single non-Gaussian source \( s \) and \( N - 1 \) uncorrelated Gaussian components, \( n \sim \mathcal{N}(0, I_{N-1}) \). We will also decompose the mixing matrix \( A \) into a column vector \( a_s \) and a \( N \times (N - 1) \) rectangular matrix \( A_n \) associated with the non-Gaussian and Gaussian components respectively,

\[
x = A \begin{bmatrix} s \\ n \end{bmatrix} = a_s s + A_n n.
\] (1)

We will consider both the on-line case, in which a new IID example \( x' \) is presented to the algorithm at each time \( t \) and then discarded, and also the batch case, in which a finite set of examples are available to the algorithm. To conform with the model assumptions the mixing matrix \( A \) must be unitary, which leads to the following constraints,

\[
\begin{bmatrix} a_s & A_n \end{bmatrix} \begin{bmatrix} a_s^T \\ A_n^T \end{bmatrix} = a_s a_s^T + A_n A_n^T = I,
\] (2)

\[
\begin{bmatrix} a_s^T \\ A_n^T \end{bmatrix} [a_s & A_n] = \begin{bmatrix} a_s^T a_s & a_s^T A_n \\ A_n^T a_s & A_n^T A_n \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & I \end{bmatrix}.
\] (3)

3 On-line learning

The goal of ICA is to find a vector \( w \) such that the projection \( y = w^T x \rightarrow s \). Defining the overlap \( R \equiv w^T a_s \) we obtain,

\[
y = w^T (a_s s + A_n n) = Rs + z \sqrt{||w||^2 - R^2} \quad \text{where} \quad z \sim \mathcal{N}(0,1),
\] (4)

where we have made use of the constraint in eqn. (2). This assumes zero correlation between \( w \) and \( x \) which is true for on-line learning but is only strictly true for the first iteration of batch learning (see section 4). In the algorithm described below we impose a normalisation constraint on \( w \) such that \( ||w|| = 1 \). In this case we see that the goal is to find \( w \) such that \( R \rightarrow \pm 1 \).
A simple Hebbian (or anti-Hebbian) learning rule was studied by Hyvärinen and Oja (1998), who showed it to have a remarkably simple stability condition. We will consider the deflationary form in which a single source is learned at one time. The algorithm is closely related to Projection Pursuit algorithms, which seek interesting projections in high-dimensional data. A typical criteria for an interesting projection is to find one which is maximally non-Gaussian in some sense. Maximising some such measure (simple examples would be skewness or kurtosis) leads to the following simple algorithm (see Hyvärinen and Oja, 1998, for details). The change in \( \mathbf{w} \) at time \( t \) is given by,

\[
\Delta \mathbf{w} = \eta \sigma \phi(y^t) \mathbf{a}^t ; \text{ followed by normalisation such that } ||\mathbf{w}|| = 1. \tag{5}
\]

Here, \( \eta \) is the learning rate and \( \phi(y) \) is some non-linear function which we will take to be at least three times differentiable. An even non-linearity, eg. \( \phi(y) = y^2 \), is appropriate for detecting asymmetric signals while a more common choice is an odd function, eg. \( \phi(y) = y^3 \) or \( \phi(y) = \tanh(y) \), which can be used to detect symmetric non-Gaussian signals. In the latter case \( \sigma \in \{-1,1\} \) has to be chosen in order to ensure stability of the correct solution, as described by Hyvärinen and Oja (1998), either adaptively or using \( \textit{à priori} \) knowledge. We set \( \sigma = 1 \) in the case of an even non-linearity. Remarkably, the same non-linearity can be used to separate both sub and super-Gaussian signals, in contrast to maximum likelihood methods for which this is typically not the case.

We can write the above algorithm as,

\[
\mathbf{w}^{t+1} = \frac{\mathbf{w}^t + \eta \sigma \phi(y^t) \mathbf{a}^t}{\sqrt{1 + 2\eta \sigma \phi(y^t) y^t + \eta^2 \phi^2(y^t)||\mathbf{a}^t||^2}}. \tag{6}
\]

For large \( N \) and \( \eta \leq O(N^{-1}) \) (two different scalings will be considered below) we can expand out to get a weight decay normalisation,

\[
\mathbf{w}^{t+1} \approx \mathbf{w}^t + \eta \sigma \phi(y^t) (\mathbf{a}^t - y^t \mathbf{w}^t) - \frac{1}{2} \eta^2 N \phi^2(y^t) \mathbf{w}^t. \tag{7}
\]

Taking the dot-product with \( \mathbf{a}_s \) gives the following update increment for the overlap \( R \),

\[
\Delta R = \eta \sigma \phi(y) \left( s^t - R^t y^t \right) - \frac{1}{2} \eta^2 N \phi^2(y^t) \mathbf{R}^t, \tag{8}
\]

where we used the constraint in eqn. (3) to set \( \mathbf{a}_s^T \mathbf{x} = 1 \). Below we calculate the mean and variance of \( \Delta R \) for two different scalings of the learning rate. Because the conditional distribution for \( y \) given \( s \) only depends on \( R \) (setting \( ||\mathbf{w}|| = 1 \) in eqn. 4) these expressions will depend only on \( R \) and statistics of the non-Gaussian source distribution.

### 3.1 Dynamics close to the initial conditions

If the entries in \( \mathbf{a}_s \) and \( \mathbf{w} \) are initially of similar order then one would expect \( R = O(N^{-\frac{2}{3}}) \). This is the typical case if we consider a random and uncorrelated choice for \( \mathbf{A} \) and the initial entries in \( \mathbf{w} \). Larger initial values of \( R \) could only be obtained with some prior knowledge of the mixing matrix which we will not assume. We will set \( r \equiv R \sqrt{N} \) in the following discussion, where \( r \) is assumed to be an \( O(1) \) quantity. The discussion below is therefore restricted to describing the dynamics close to the initial conditions. For an account of the transient dynamics far from the initial conditions and the asymptotic dynamics close to an optimal solution, see Rattray (2002).

#### 3.1.1 \( \phi(y) \) even, \( \kappa_0 \neq 0 \)

If the signal is asymmetrical then an even non-linearity can be used, for example \( \phi(y) = y^2 \) is a common choice. In this case the appropriate (ie. maximal) scaling for the learning rate is \( O(N^{-\frac{2}{3}}) \) and we set \( \eta = \nu / N^{\frac{2}{3}} \) where \( \nu \) is an \( O(1) \) scaled learning rate parameter. In