Project in Statistics - ICA

Philip von Brockdorff - brs604
supervised by Sebastian Weichwald

June 12, 2020

Abstract

Finding a suitable representation of large scale multivariate data can be a troublesome task, even more so when the data contains much noise. PCA may serve as tool for whitening the data, but often this is not enough. Sometimes it will seem natural to assume an underlying structure of the observed data; namely that the data is generated from some unobserved sources. Finding a representation for these sources is known as the blind source separation problem. ICA can be used as a tool for this problem. We will in this project consider several linear ICA models and employ them on an EEG-dataset provided by [6]. We will use a pipeline similar to that of [4], we will compare results, and we will test various steps of our pipeline. While our ICA models perform worse than those of [4], we observe better results (which are on par with the best of [4]) using a simple model. We view these results as an illustration of the difficulties of working in a large scale unsupervised setting, and as reminder to use more measures to judge performance.

1 Introduction

Suppose that we are in a room with two people speaking simultaneously, and have two microphones recording, held at different locations. The two microphones give us recorded time signals, let us denote them by $x_1(t), x_2(t)$ for every time index t. Each of these recordings is a weighted sum of the two voices. Let us denote the voices by $s_1(t), s_2(t)$. This gives rise to the linear equation

$$\begin{aligned} x_1(t) &= a_{11}s_1(t) + a_{12}s_2(t) \\ x_2(t) &= a_{21}s_1(t) + a_{22}s_2(t) \end{aligned}$$

where the a parameters could describe the distance of the speaker to the microphone. The task is here to estimate the source signals s_1, s_2 having only observed the recordings x_1, x_2. This is called *the Cocktail Party Problem* and it turns out that we may indeed estimate the source signals s_1, s_2 (so as to distinguish one voice from the other), if they are independent at each time index t. ICA models can be used to estimate the a parameters, as well as the source signals s_1, s_2.

The problem above can be written in matrix form instead, so that $X(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}, S(t) = \begin{pmatrix} s_1(t) \\ s_2(t) \end{pmatrix}$, and

$$X(t) = AS(t), \text{ where } A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}.$$

Illustrative example

Assume that we have observed the signals $X(t)$ in Figure 2 with $X(t) \in \mathbb{R}^2$ for each $t \in \{1, .., 500\}$. Figure 1 shows the original source. Using ICA we were able to recover the sources in Figure 3 only having observed the mixed signals $X(t)$. Note that these only differ from the original sources by a sign - see *Ambiguities of ICA* in Section 2.

America's Got Talent Duet Problem

The two previous examples were not completely realistic. In the real world there will almost always be some noise to any observed data. In the cocktail party problem this may be included by adding a noise vector, we will later define the noisy ICA model formally.

For now we instead consider an extended version of the cocktail problem. Here we have a pair of microphones, and a pair of singers at a duet audition. Further noise is added to this problem, namely an audience and two open windows. This is called *the America's Got Talent Duet Problem* and the task lies in estimating each singers voice.[1]

[1] An example with artificial sound data can be seen on https://sweichwald.de/coroICA/.

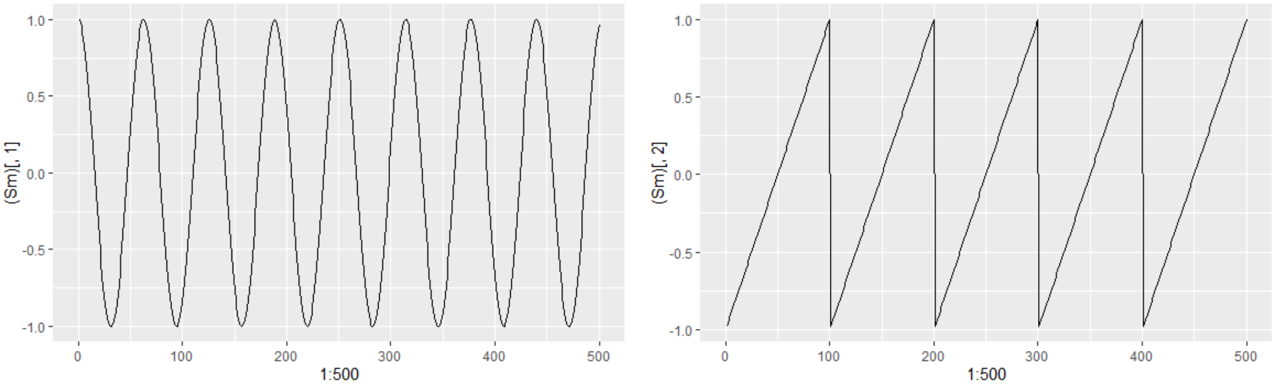

Figure 1: Sources for signals

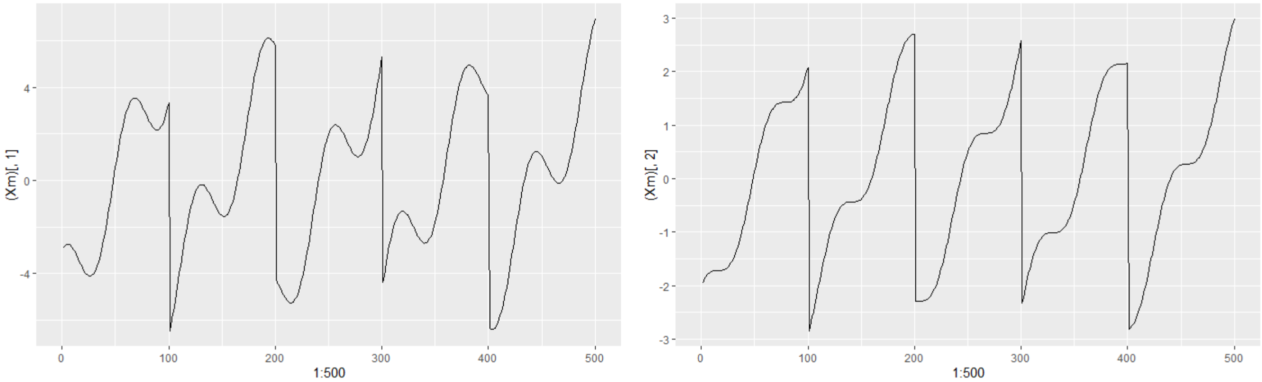

Figure 2: Observed mixed signals

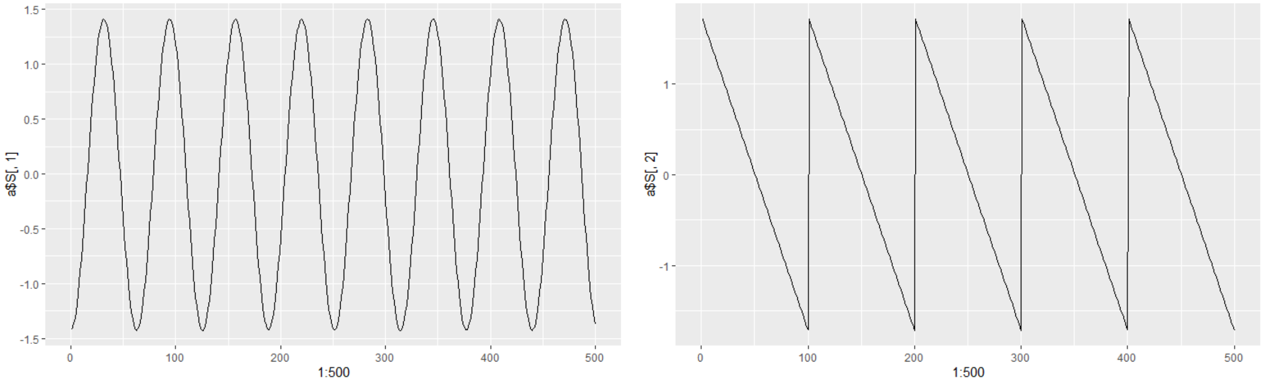

Figure 3: Recovered sources by fastICA

This setting contains more noise than the previous cocktail problem. We will later define the group-wise noisy ICA model, which is appropriate for this setting.

Further uses of ICA

There are many applications of ICA, specifically as a tool to deal with the blind source separation problem. Beyond the audio settings above, ICA can also be used in an image setting, financial setting and for brain imaging applications. See part IV of [1].
It is the latter setting we will concern ourselves with in this project. Electroencephalograms (EEG) are recordings of electric fields of signals emerging from neural currents within the brain. They are recorded using electrodes placed on the scalp.

2 Methods/theory

Assume that we have observed $X_1, ..., X_n$ with each $X_i \in \mathbb{R}^d$. The classic ICA (Independent Component Analysis) model is defined as

$$X_i = AS_i, \quad \forall i \in (1, .., n). \tag{1}$$

Where each $X_i \in \mathbb{R}^d$ is a random variable of observed mixed signals, $A \in \mathbb{R}^{d \times d}$ an invertible mixing matrix and $S_i \in \mathbb{R}^d$ a random variable of components/signals.
We assume that each X_i is a linear mixture of the underlying components of the signal S_i, which are latent variables i.e. unobserved. We further assume that the components are mutually independent. That is,

$$\text{if } S_i = \begin{pmatrix} s_{i_1} \\ s_{i_2} \\ \vdots \\ s_{i_d} \end{pmatrix}, \text{ then } s_{i_j} \perp\!\!\!\perp s_{i_k}, \forall j \neq k, \quad \forall i \in \{1, .., n\}.$$

This assumption need not be completely true in practice. Indeed we will later see that the best performance is achieved when this is not the case.

Furthermore, the independent components (IC's) must have non-Gaussian distributions (see Section 7.5 in [1]).

Above we defined A to be square. This is the case when we assume that the amount of IC's is the same as the amount of observed mixed signals. This need not be the case and the ICA model does also work in the case where we assume less IC's than mixtures, here A has a left-inverse instead of being invertible. For the cocktail party problem this setup would correspond to having more microphones than people talking. We will not consider the case of more IC's than observed signals.

Ambiguities of ICA

For now we drop the sample index i for readability. In essence the aim of ICA is to extract the signals S, and maxing matrix A, while having only observed X. This is known as the blind source problem and may in theory be troublesome: For any invertible matrix P, we have

$$AS = AIS = AP^{-1}PS.$$

So we cannot identify the true underlying signals; that is, whether the observed mixed signals X are generated by PS or S. Indeed as P can be any invertible matrix we cannot tell the order (e.g. if P is a permutation matrix), nor the power/variance of the true signals (any scaling of the source signal may be canceled by dividing the corresponding row of P). The former will not matter in most cases. While the latter may somewhat be circumvented by assuming the signals have unit variance. Note that the sign of the signal will still not be identifiable (see Figure 1 and Figure 3). This is not a problem in most cases either; for instance in our case we take the variance of the signal.

Hence the mixing matrix is uniquely identifiable up to a scalar (for proof see [9, 10]).

Pre-processing data

Thus, without loss of generality we may assume that the signals and observed mixed signals have one variance. We may further assume that they are centered, i.e. have zero mean (section 7.2.4 [1]).

Whitening the data before doing ICA is often recommended. When we do so we will use PCA (Principal Component Analysis). PCA and ICA are closely related. Both may be used for whitening the data and are indeed used for high-dimensional data. While ICA assumes an underlying independent composition generating the data and attempts to solve the blind source separation problem, PCA seeks to explain the data with (fewer) uncorrelated components instead.

Typically when we use PCA, we choose the amount of components by how much variance they can explain; by considering the eigenvalues for the covariance matrix. Furthermore, the components are ordered by how much variance they can explain. When we use ICA all components are equally important.

Finally while PCA uses 2nd order statistics, ICA considers higher order statistics such as the kurtosis.

ICA variations

There exist many variations of ICA. We will introduce the time-series structured ICA formally now.

The independent components of the classic ICA model may be assumed time signals. In this case the sample index i may be replaced by the time index t, which defines an ordering between the signals. Specifically, the IC's depend on the time index.

$$X(t) = AS(t), \quad \forall t \in \{1, .., n\}. \tag{2}$$

As before, $X(t), S(t) \in \mathbb{R}^d, A \in \mathbb{R}^{d \times d} \quad \forall t \in \{1..., n\}$.

In this model the non-Guassianity assumption of the IC's is replaced by assumptions regarding the time-structure: That is, that the different IC's have different autocovariances[2], all different from zero or that the variances are nonstationary [3] (See chapter 18 [1]).

Noisy ICA

We can extend the classic ICA model to the noisy ICA model by adding a noise term

$$X_i = AS_i + N_i, \quad N_i \in \mathbb{R}^d, \quad \forall i \in \{1, .., n\}. \tag{3}$$

Here we further assume that the noise is independent from the independent components and that the noise is Gaussian. Under these assumptions the mixing matrix is identifiable [3], however the realizations of the independent components may no longer be identified, since they cannot be completely separated from noise.

If we use the time-series approach instead, replacing the sample index i with the ordered time index t in 3 we get

$$X(t) = AS(t) + N(t), \quad \forall t \in \{1, .., n\}. \tag{4}$$

Groupwise confounding noisy ICA

We will now extend the noisy ICA model 3 to a groupwise model. For completeness, we will also present the assumptions needed to prove identifiability of the mixing matrix [4]. We will make assumptions about the structure of the source signals and noise.
Assuming we have observed $X_1, .., X_n$ consider

$$X_i = AS_i + H_i, \quad \forall i \in \{1, ..., n\}. \tag{5}$$

With $X_i, S_i, H_i \in \mathbb{R}^d, A \in \mathbb{R}^{d \times d}$, S_i and H_i independent. Here H_i are stationary confounding noise variables with fixed covariance within each group. We will follow the definition of [4] and make this model correspond to the *variance signal.* In essence there still is some time-dependencies, as the variance process of the signals is allowed to change over time.

Here the i is the sample index, but we may as well replace this with the time index t so that

$$X(t) = AS(t) + H(t), \quad \forall t \in \{1, .., n\}. \tag{6}$$

This model will correspond the other signal variant of [4], that is the *time-dependence signal.* Here the time-dependence may change over time.

Furthermore, we assume the existence of some underlying group structure for the confounding matrix. That is, that the signals may be partitioned into groups: Formally that there exists a group $\mathcal{G}$ with

$$\mathcal{G} = \{g_1, .., g_m\}, \text{ where } \quad g_k \subseteq \{1, .., n\}, \quad \bigcup_{k=1}^{m} g_k = \{1, .., n\}$$
$$\text{such that } (H_i)_{i \in g} \text{ is weakly stationary } \forall g \in G.$$

[2] cov(S(t_i),S(t_j)), $\quad i \neq j$

[3] The variances are time varying.

So that the noise is allowed to vary from group to group, but is weakly stationary in each group. This is Assumption 1 of [4].

Thus we may now think of the observations $X_1, .., X_n$ as being generated from some signals S_i and noise H_i which have some underlying group mechanic.

Assuming Model 5, we further assume that for each pair of components $i, j \in \{1, .., d\}$, there exist $g_1, g_2, g_3 \in \mathcal{G}$, not necessarily unique, with $l_1, k_1 \in g_1, l_2, k_2 \in g_2, l_3, k_3 \in g_3$ such that

$$\begin{pmatrix} \mathrm{Var}(S^i_{l_1}) - \mathrm{Var}(S^i_{k_1}) \\ \mathrm{Var}(S^i_{l_2}) - \mathrm{Var}(S^i_{k_2}) \\ \mathrm{Var}(S^i_{l_3}) - \mathrm{Var}(S^i_{k_3}) \end{pmatrix} \text{ and } \begin{pmatrix} \mathrm{Var}(S^j_{l_1}) - \mathrm{Var}(S^j_{k_1}) \\ \mathrm{Var}(S^j_{l_2}) - \mathrm{Var}(S^j_{k_2}) \\ \mathrm{Var}(S^j_{l_3}) - \mathrm{Var}(S^j_{k_3}) \end{pmatrix}$$

are neither colinear nor equal to zero. This is Assumption 2 of [4].

Assuming Model 6, we instead assume that for each pair of components $i, j \in \{1, .., d\}$, there exists $g_1, g_2, g_3 \in \mathcal{G}$, not necessarily unique, with $l_1, k_1 \in g_1, l_2, k_2 \in g_2, l_3, k_3 \in g_3$ for which there exists $\tau \in \{1, .., n\}$ such that

$$\begin{pmatrix} \mathrm{Cov}(S^i_{l_1}, S^i_{l_1-\tau}) - \mathrm{Cov}(S^i_{k_1}, S^i_{k_1-\tau}) \\ \mathrm{Cov}(S^i_{l_2}, S^i_{l_2-\tau}) - \mathrm{Cov}(S^i_{k_2}, S^i_{k_2-\tau}) \\ \mathrm{Cov}(S^i_{l_3}, S^i_{l_3-\tau}) - \mathrm{Cov}(S^i_{k_3}, S^i_{k_3-\tau}) \end{pmatrix} \text{ and } \begin{pmatrix} \mathrm{Cov}(S^j_{l_1}, S^j_{l_1-\tau}) - \mathrm{Cov}(S^j_{k_1}, S^j_{k_1-\tau}) \\ \mathrm{Cov}(S^j_{l_2}, S^j_{l_2-\tau}) - \mathrm{Cov}(S^j_{k_2}, S^j_{k_2-\tau}) \\ \mathrm{Cov}(S^j_{l_3}, S^j_{l_3-\tau}) - \mathrm{Cov}(S^j_{k_3}, S^j_{k_3-\tau}) \end{pmatrix}$$

are neither colinear nor equal to zero. This is assumption 3 of [4].

Now assuming either Assumption 1 and 2, or Assumption 1 and 3, the mixing matrix A is identifiable (Theorem 1 of [4]).

We have now presented several ICA moodels. There exists many others, e.g. non-linear versions, but for this project the ones presented will suffice. In practice it may be the challenging to identify which model to assume. For instance, real world data will almost always have some underlying noise, but this does not make the noiseless ICA models inapt.

We will now present algorithms that deal with some of these models.

FastICA

For a random variable y with density $f(y)$, the entropy is defined as

$$H(y) = -\int f(y) \log f(y) dy.$$

Since *a Gaussian variable has the largest entropy among all random variables of equal variance* ([2, 11]) we may use entropy as a measure of non-Gaussianity. The negentropy J may be defined as

$$J(y) = H(y_{Gauss}) - H(y),$$

where y_{Gauss} is a Guassian random variable of the same covariance matrix as y. The negentropy may be approximated by

$$J(y) \approx \frac{1}{12}\mathrm{E}[y^3]^2 + \frac{1}{48}\mathrm{kurt}(y)^2,$$

or, if y has a symmetric distribution,

$$J(y) \approx (\mathrm{E}[G(y)] - \mathrm{E}[G(v)])^2, \quad v \sim N(0,1),$$

for almost any non-quadratic function G. It turns out that using either $G(u) = \frac{1}{\alpha}\log(\cosh(\alpha u))$ (with differential $g(u) = \tanh(\alpha u)$) for $1 \leq \alpha \leq 2$ or $G(u) = -\exp(-\frac{u^2}{2})$ (with differential $g(u) = u\exp(-\frac{u^2}{2})$) provide good results.

Hence maximizing the negentropy will maximize non-Gaussianity.

FastICA assumes the classic ICA model 1. FastICA whitens the data matrix X with PCA so that KX will be the principal components. Then the algorithm estimates the unmatrix W so that $WKX = S$. The matrix W is estimated by maximizing the approximated negentropy under the constraint that W is orthonormal. (See chapter 8 for a more thorough discussion [1] or [2].)

While FastICA assumes the noiseless ICA model, it has shown to do well on noisy data as well.

CoroICA

A different approach to ICA is given by tensorial methods, these use (second- and fourth-order) cumulant tensors. See chapter 11 of [1] for details. Said chapter presents two main approaches:

- FOBI (fourth-order blind identification) - which attempts to jointly diagonalize the covariance matrix and fourth-order cumulant matrix.
- JADE (joint approximate diagonalization of eigenmatrices) - which attemps to diagonalize several different fourth-order cumulant matrices.

Which both assume non-Gaussian signals. Note that generalizing FOBI leads to JADE.

Now we instead assume the time-series structure, with the signals having a weakly stationary structure. This model is often referred to as the second-order source-separation model (SOS). This approach gives rise to methods such as AMUSE, SOBI, choiICA and ultimately coroICA.

These methods differ in which matrices (whether it be covariance, autocovariance or both) they jointly diagonalize and how they do so. And also, which kind of noise they allow.

- AMUSE - diagonalizes the covariance matrix and the auto-covariance matrix for one fixed lag.
- SOBI- often used in EEG data. Uses all lags up to a certain order and jointly diagonalizes all the resulting auto-covariance matrices.
- ChoiICA- attemps to jointly diagonalize blocks of covariances, auto-covariances, or covariances and auto-covariances.

CoroICA (confounding-robust ICA) is an algorithm introduced and developed in [4]. Assuming the group-wise confounding noisy model, i.e. either Model 5 or 6, we may estimate the unmixing matrix by using the group structure. Specifically in the following way:

- Partitioning groups into subgroups.
- Compute emperical covariances (or auto-covariance if we assume the time-series structure) on each subgroup.
- Estimate a matrix that simultaneously diagonalizes the difference of these emperical (auto-)covariance matrices using an approximate joint matrix diagonalization technique.

We note that coroICA can also be used if we assume no group-wise confounding noise. In this case we simply have one group.

3 Data

Dataset

We will investigate the BCI IV competition dataset 2b *4-class motor imagery* [6, 7]. This dataset consists of 18 sessions of 9 subjects each with 2 sessions on different days. The recordings are cue-based with four different motor imagery tasks, namely the imagination of movement of the left hand (class 1), right hand (class 2), both feet (class 3), and tongue (class 4).

The recordings are from 22 EEG channels and 3 EOG channels, with electrodes placed as illustrated on Figure 4.

Each session consists of 6 runs, each with 48 trials (evenly distributed, so that there are 12 trials for each task), yielding a total of 288 trials per session.

Prior to the 6 runs a recording of approximately 5 minutes was performed, divided into 3 blocks: One with eyes open, one with eyes closed and one with eye movement. As seen in Figure 5 (left).

Each trial is 6 seconds long followed by a short break. At the start of a trial (t=0s) a fixation cross appeared, as well as a short acoustic warning tone. After two seconds a cue in the form of an arrow corresponding to which of the four tasks to perform was shown for 1.25s. At (t=3s to t=6s) the subject performed the desired motor imagery task. No feedback was provided. Figure 5 (right) illustrates this timing scheme of the paradigm.

22 Ag/AgCl electrodes (with inner-electrode distance of 3.5 cm) were used to record the EEG. The signals were sampled at 250 Hz and bandpass-filtered between 0.5 Hz and 100 Hz. The sensitivity of the amplifier was set to 100 μV. An additional 50 Hz notch filter was enabled to suppress line noise. The 3 EOG channels instead had the amplifier set to 1 mV.

Finally, a visual inspection of all data sets was carried out by an expert and trials containing artifacts were marked.

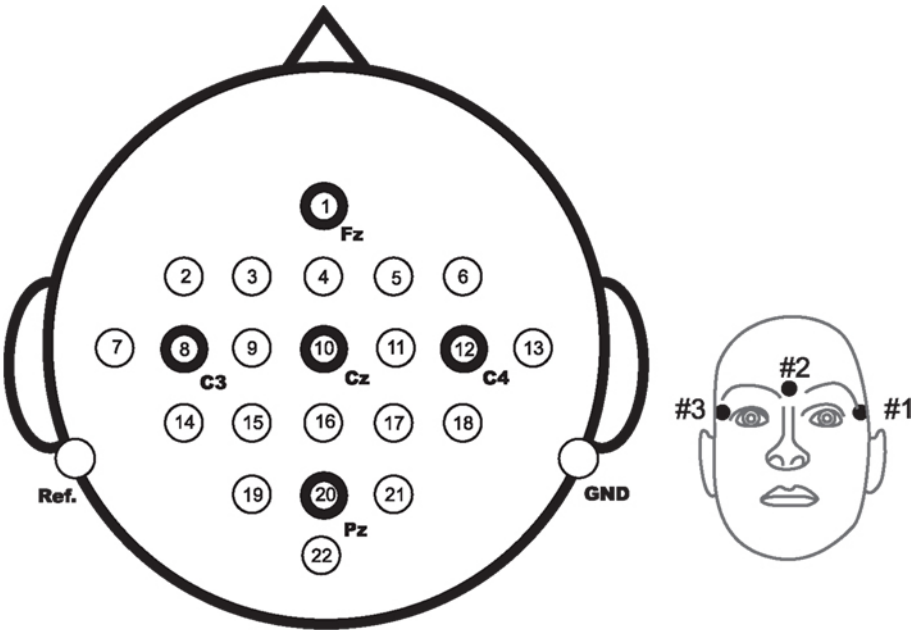

Figure 4: Electrode placement for the dataset BCI IV 2a. As seen in [6, 7].

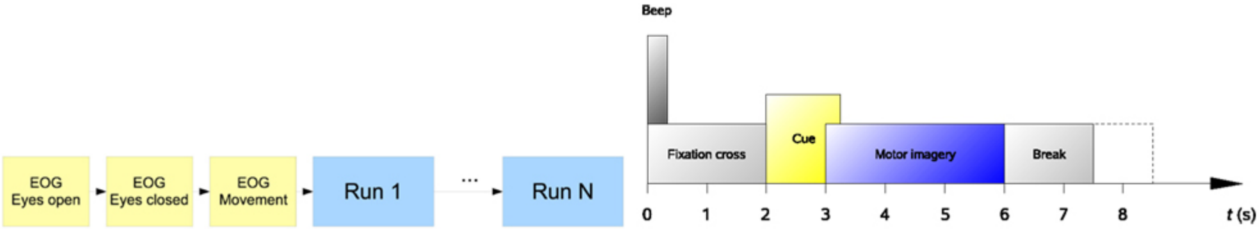

Figure 5: Session and trial visualisation for the dataset Data for BCI IV 2a. As seen in [6, 7].

Details

There are many ways to investigate the data.

In the competition setting only labels for the first session of each subject were given and the aim was to predict the labels of the second session.[4]

We will consider this setting as a training data vs test data setting; where the training data is the first session of all subjects and the testing data the second session off all subjects. Here we are comparing results across sessions for all subjects.

The remaining labels were released after the competition ended.

In [4] the performance of various different ICA models was instead evaluated by using a leave-k-subjects-out (k=1,..,8) fashion in the following sense: Using all but k subjects to train a model, and this model is then used to predict the trials of the remaining k held-out subjects.

This setting varies from the competition setting in the following sense: Here the training data is both sessions for all but k subjects, and the test data is both sessions for the remaining k subjects. Here we are comparing results across subjects for all sessions.

We employed a mixture of these two settings.[5] For computational and time restrictions, we will only be using the first session of each subject for model building. Specifically in the following sense: We used only the first session of all subjects with the same leave-k-subjects-out approach, but for fixed $k = 1$. Thus we will train a model on 8 subjects and test the performance on the last held out subject. We do this for each subject and use the average of these performances as our measure of performance. We refer to this as the test-accuracy. This approach is comparable to Figure 16 of [4] with 8 training subjects, the only difference being that in the article they use both sessions.

We will only use parts of the data set. Specifically we will use the 22 EEG-channels, the 6 runs and 3 second bits of trials (from t=3s to t=6s) - as was done in [4].

4 Training protocol

Some key parts of the process of classifying EEG data are:

1. Data-preprocessing.
2. Feature extraction and selection.
3. Classification.

Our analysis will focus on the first step.
We will follow the protocol used in [4] with some alterations. Most notable differences are that:

- We only use half the data files.
- We apply centering, scaling and PCA, instead of CAR and projecting onto the orthogonal null-space.
- At times we will CAR the extracted features.
- We use a simpler classifier.
- We have done the implementations in R instead of Python.

We will investigate how various steps (and parameters of these steps) may influence the performance. We will apply fastICA and coroICA to compare the two. We will use the bandpower of the signals as features to classify with a shrinkage Linear Discriminant Analysis (sLDA) classifier.

We note that while [4] uses several measures (measure minimum distance (MD), mean covariance instability score (MCIS) and classification accuracy) for many ICA models (coroICA,fastICA,choiICA,SOBI) on all possible combinations of training-test subjects, we will instead only measure classification testing accuracy for coroICA and fastICA with 8 training subjects.

[4] The contestants were judged using a κ measure, and the winner used an algorithm based on the filter bank common spatial pattern (FBCSP) variant[8].

[5] Or perhaps, a simpler version of the latter.

Pipeline

We sketched our approach for model building above. For clarity we now present in details how we approached the problem.

For each subject $S_i, i = 1, .., 9$ we may denote the data matrix with the 22 signal recordings as X_{S_i} and Y_{S_i} as the trial labels.

Since the signals are sampled at 250 Hz, and each session has $6 \cdot 48$ trials (each of 3 second length), the resulting dimensionality of X_{S_i} is $6 \cdot 48 \cdot 3 \cdot 250 \times 22 = 216000 \times 22$, and likewise Y_{S_i} is a vector of length $6 \cdot 48 = 288$

$$X_{S_i} \in \mathbb{R}^{216000 \times 22}, \quad Y_{S_i} \in \mathbb{R}^{288}, \quad \forall i \in \{1, .., 9\}.$$

Where the first trial is the first 750 rows of X_{S_i}, whose label is first element of Y_{S_i}.

Now let $k \in \{1, .., n\}$ be the index of the left out subject. We now define the training- and test set as

$$X_{Train} = \bigcup_{i \neq k} X_{S_i}, \quad Y_{Train} = \bigcup_{i \neq k} Y_{S_i}$$

$$X_{Test} = X_{S_k}, \quad Y_{Test} = Y_{S_k}.$$

The union sign here meaning that we append the matrices by rows. So the resulting training set X_{Train} is of dimensionality $8 \cdot 216000 \times 22$, with labels $Y_{Train} \in \mathbb{R}^{8 \cdot 288}$.

Now given X_{Train}, X_{Test} we follow the protocol presented in Table 2. We do this 9 times, once for each subject left out, and take the average of the test-accuracies.

Note that we are taking the bandpower of the corresponding trials. For example, trial 1 of the training set corresponds to the first 750 rows of S_{Train}. Taking the diagonal of the corresponding covariance matrix (of dimension $d \times d$) yields a vector of length d. We use the log of this as a feature to classify.
Here d is the amount of signals we use, which is 22 if we do not whiten the data. Using this approach we thus reduced the task of classifying one trial, which is a 750×22 matrix, to the task of classifying one feature for a trial, which is a vector of length d.

It is worthy to note that when we define the training set as the union of the subjects some of the time-series assumptions may seem inapt if we were to apply corresponding ICA models. Luckily, with coroICA we may partition the groups by subject, thus keeping the time-series assumptions valid. On the contrary, fastICA assumes no time-series structure, but instead non-Gaussianity.

5 Results

We present the test-accuracy for 1 subject left out. That is, we train a model on 8 subjects and test it on the the last. We did this for all subjects and took the average of the test scores. For classifying we used a 200 bootstraped sLDA classifier. We present the following three main results, with the results in [4] as comparison.:

- Using fastICA we were able to achieve 37% test-accuracy on average at best (about 38% in [4]).
- With coroICA we were only able to achieve 36% test-accuracy on average at best (about 39% in [4]).
- Finally, using a simple model, which uses no ICA nor PCA, we managed to get 39.3% testing accuracy.

Note that [4] did these results on all the data (both sessions for each subject), while we focused on the first session of each subject.

We were able to achieve decent results even without many of the protocol steps. For instance:

- When not using ICA (but using PCA with either N=22,21,14) we got $29 - 30\%$ test-accuracy.
- Not using PCA nor ICA we were able to achieve test-accuracy of 39.3%.
- Not using filtering either gave us a test-accuracy of 37.4%.
- Not using the above nor common average reference (CAR) gave us a test-accuracy of 34.8%.
- And with no centering nor scaling we achieved test-accuracy of 33.2%.

At the final step we are only taking the bandpower of the raw 3-second trials.

We have some interesting observations here. First off, we see that PCA on its own (that is, without ICA) does not perform well. The results are above chance-level, but even the raw 3-second trials outperform it. Indeed PCA does not handle the blind source separation problem at all. Perhaps instead some of the signals are mixed/entangled further (as compared to unmixed doing ICA), making it harder to classify.

We see that many of the steps do prove to be useful.

With these results we may be compelled to dismiss the PCA and ICA step, so let us investigate this model. We will refer to this model as the simple model.

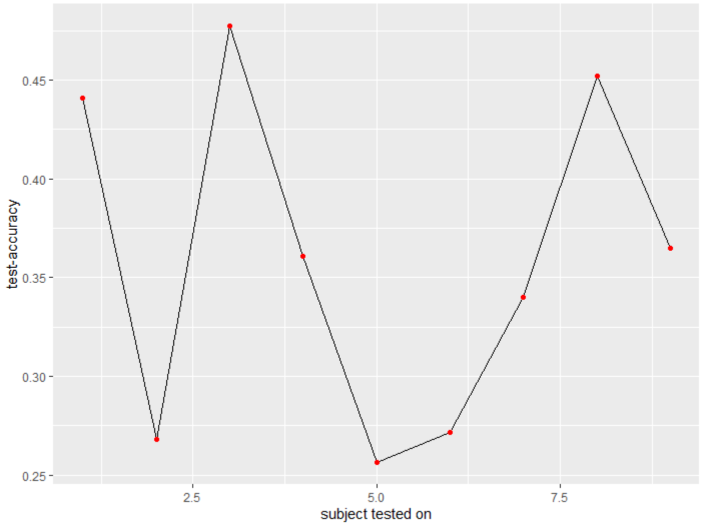

Figure 6: Varrying test-accuracy according to subject left out using fastICA with N=14 comps.

Simple model

We would expect the ICA models to do better on noisier data (coroICA by its nature, fastICA by robustness in other papers [2, 3]).

One way of ensuring the data is more noisy is by the following approach:

- Removing artifacted trials from the training, while keeping artifacted trials in the test set.

This approach should ensure that the model is more likely to train on brain-signals rather than muscle movement. In which case the performance of the ICA models should not suffer by a lot.

Indeed, both ICA models produce similar results when using this artifact free training set approach. Interestingly the simple model also does manage so; achieving test-accuracy of 38.9%. We note that we also did filter when doing this. It could prove worthwhile to remove the filtering step in addition to the above, but we did not find the time to do so.

Further testing with regards to this simpler model would be appropriate. Especially training-accuracy (eg. cross-validation) and comparing this to the training-accuracy of ICA models.

In any case, the results we achieved with this simple model were on par with the best results achieved in [4].

FastICA

The best results for fastICA were achieved when we did not use CAR, giving us test-accuracy of 37%.

Figure 6 illustrates the trouble of classifying according to which subject is left out. We especially see how fastICA does well on some subjects and less so on other.

Figure 7 illustrates the performance as a function of components used. The best score is at N=14 with 36.7% test-accuracy. We tested performance with CAR:

- Using CAR at step 11 and 12 in Table 2 we achieved 35.7%, 34.2% test-accuracy with N=14, N=22 components respectively.
- Not using CAR at step 11 and 12 instead gave 37%, 36.1% test-accuracy with N=14, N=22 components respectively.
- Not using CAR, nor centering, nor scaling dropped the performance to 34.7% with N=14 components.

We tested the performance given ordering of filtering:

- Compared to the best performance above, 37% when filtering after applying fastICA.
- We only achieved 33.3% using filtering before applying fastICA.

The approach of using filtering as a preprocessing tool before applying ICA is suggested in Chapter 13 [1].

Furthermore, using CAR instead of centering and scaling, and removing the CAR step (step 11 and 12 of Table 2) gave 35.8%, 34% test-accuracy with $N = 14, N = 21$ components.

Finally, note that when using PCA, testing show that N=20, N=9, and N=5 components correspond to 99.9%, 99%, and 95% explained variance respectively.

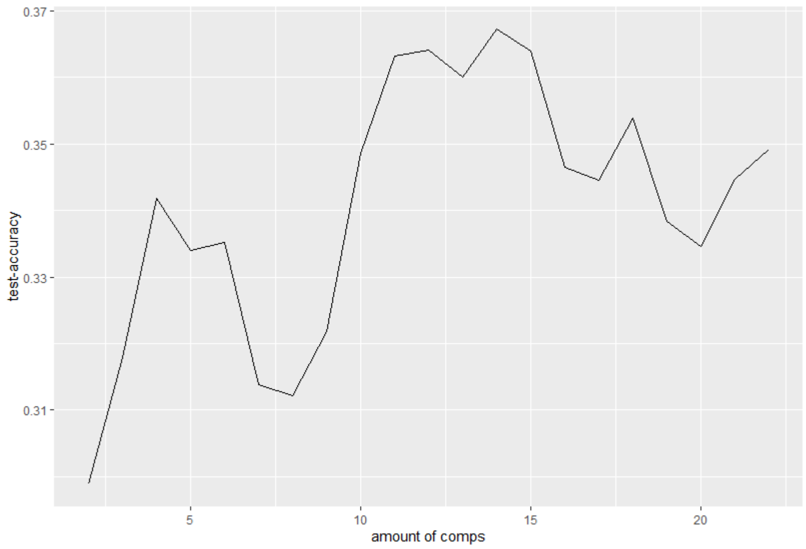

Figure 7: Test-accuracy as a function of N-components for fastICA with N=2,..,22.

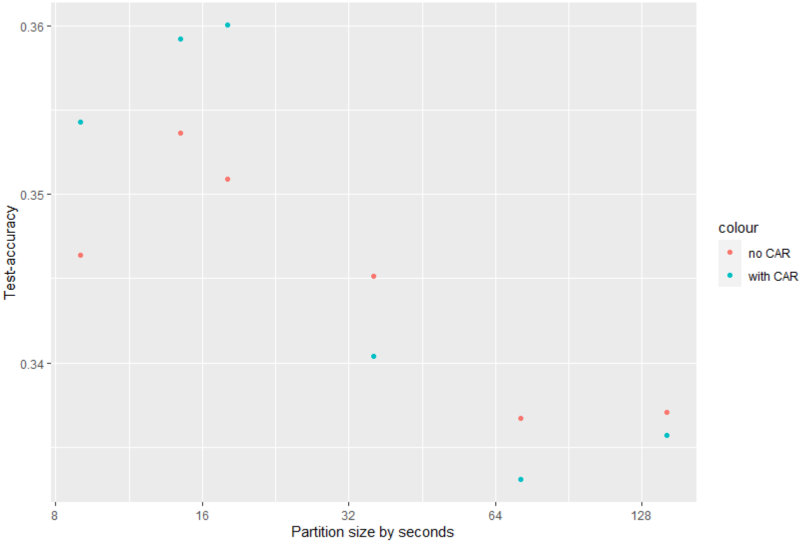

Figure 8: Test-accuracy for coroICA, given partitionsize with/without CAR.

CoroICA

The best results for coroICA were achieved when we did not use PCA, giving us test-accuracy of 36%.

Figure 8 illustrates the performance of varying partition size. We especially see that the 15 second partition in [4] is sensible. Here CAR has been applied (or not applied) at step 11 and 12 of Table 2. The best results are achieved with CAR, however not using CAR may hint to be more robust.
We also attempted to follow the pipeline in [4] more closely; by applying CAR instead of centering and scaling and removing the later CAR step. This ensures that one of the 22 signals is a linear combination of the others, hence whitening the data may seem safe. We applied PCA with 21 dimensions and then used coroICA. However, this gave worse results, with test-accuracy of 35%. We will later attempt this on the full data set and see good results.

The order of when we filter also matters, as it did for fastICA. If we filter before using coroICA, instead of after using coroICA, the test-accuracy drops to 32.5%.

Full dataset

For completeness we will also present the performance of the three models on the full data set (that is, including the second session). This setting is the same as the one in [4]. We note that we did not focus on this setting, but merely applied the models that seemed to do well in our main setting.

Both the simple model and fastICA will be used with the same settings as performed best on the smaller

Model	Best performance on half the dataset	Performance on full dataset	Difference in performance
Simple model	39%	38%	-1
fastICA	37%	34.5%	-2.5
coroICA	36%	37% (37.7%)	+1 (+1.7)

Table 1: Performance (test-accuracy) given the best setting for each model, with half the dataset and the full dataset.

data set. While for coroICA we will use two different settings, the first being the one which did the best on the smaller data set. The second mimicking a closer pipeline to the one of [4].

On Table 1 we see that coroICA is the only model which improves in test-accuracy given more data. We see that fastICA suffer the most.

Had we instead focused on this setting we may have seen closer performance to that of [4]. We were able to achieve 37.7% test-accuracy using the [4] approach of using CAR+PCA(N=21).

Indeed the increased performance of coroICA is promising and would have been interesting to test further. Likewise the drop in performance of fastICA is interesting, perhaps using only 14 compositions is hurting the accuracy here.

6 Discussion

We have now looked at several models and their performance. All models were above chance-level.

We have seen interesting results. On the smaller data set fastICA did better than coroICA, while coroICA outperformed fastICA on the bigger data set. However the simple model outperformed both.

Theory

The assumption of all the presented ICA models is that there exist some underlying independent signals which (to some extend) generate the observed data. In the case of EEG data, some of those underlying signals would be brainwaves, and the ICA model would aim to present these (along with other signals, which would account for some of the noise).

This assumption is reasonable; however extracting the signals, and knowing (or guessing) how many there are, is a hard task. Especially as the data is so noisy. Furthermore, we cannot check if the signals we extract are actually signals corresponding to the brainwaves. [6]

With fastICA we achieved best performance using only 14 compositions, corresponding to the assumption that there are 14 underlying signals. Prior to the experiments we had no theoretical ground for why 14 (or any other number of) signals may do well. On Figure 7 we see that $N = 4$ compositions does reasonably well too, but clearly some of the complexity is not captured without more compositions.

CoroICA assumes some underlying group structure. Logically this assumption should hold for the data; each session is different from another:

- It is logical to expect some variance across different subjects.
- But also some variance for different sessions for a fixed subject, such as variance of the equipment (placement of the electrode-cap), variance of the subject, variance of external factors (background waves).

Figure 6 supports the former claim, and results from the full data set support the latter.

With the simple model we achieved better results. This model makes no assumptions, nor does it aim to extract any signals.

Nonetheless, ICA methods have proved to do well with EEG data.

Results and Classifier

We will now further discuss some of the results observed.

Filtering

We observed drop of accuracy when we filtered before applying ICA.

When we apply the 8-30 Hz bandpass filter we are removing some of the noise of the data.

- When we apply the filter prior to ICA, the ICA methods instead attempt to separate less noisy data. This may cause the ICA methods to disregard this noise.

[6] Note that [4] presents several measures, also some to measure the strength of the signals.

- When we instead do not apply the filter prior to ICA, the ICA methods may (correctly) estimate this noise as being created by some source signal. And may thus specify this signal as being a generator of the noise. Then when we apply the filter after ICA, we remove the very noisy signals.

The latter approach may make for an easier (less-noisy) classification. We observed better results using this approach.

Test-accuracy

The only measure we used was test-accuracy and as previously mentioned, using more measures would be apt, as was done in [4].

Judging purely by test-accuracy we have seen how even a simple model may outperform more complex ICA model.

If we were in a competition setting, where we primarily aim for high test-accuracy, we would have to focus much more on the classifier. For instance, the amount of motor-imagery tasks of each session was evenly split so you could incorporate this information in your classifier. Indeed, a classifier only able to distinguish between two of the imagery tasks could achieve 50% accuracy in this case.

Furthermore, a very complex classifier may incorporate all the noise of the data to correctly classify, while disregarding any brain-signal based observations. Instead we employed a simple linear classifier, which ensures the ICA model linearity and allows us to compare results.

CoroICA only outperformed fastICA when we included all the data. Suggesting that coroICA is able to do well with larger data. The increased performance in this case is fascinating. Especially considering that the pipeline had to be modified to achieve those results. I.e. different sized data requires different pipelines to perform well. The strong performance of the simple model is interesting as well. In essence this illustrates some of the difficulties of working in a large unsupervised (noisy) setting. There is no ”cookie-cutter” approach, and indeed many cogwheels to consider.

Appendix

1. Centering and scaling X_{Train}.
2. Centering and scaling X_{Test}.
3. Find PCA composition for X_{Train}, and K, so that $PCA_{Train} = X_{Train}K$.
4. Find PCA composition for X_{Test} using the K found above, i.e., $PCA_{Test} = X_{test}K$.
5. Apply ICA on PCA_{Train} resulting in signals S_{Train} and unmixing matrix V.
6. Find signals for the test set using the unmixing matrix above, that is, $S_{test} = PCA_{Test}V$.
7. Filter S_{Train} with a 8-30 Hz bandpass-filter.
8. Filter S_{Test} with a 8-30 hz bandpass-filter.
9. Take the bandpower of S_{Train}, i.e. $Fe_{Train} = \log(diag(Var(S_{Train})))$.
10. Take the bandpower of S_{Test}, i.e. $Fe_{Train} = \log(diag(Var(S_{Test})))$.
11. CAR Fe_{Train}.
12. CAR Fe_{Test}.
13. Classify S_{Test} and compute test-accuracy.

Table 2: Detailed pipeline for testing on one subject. Doing this over all the subjects gives the reported test-accuracies.

References

[1] A. Hyvärinen, J. Karhunen, E. Oja.
Independent Component Analysis. A Wiley-Interscience Publication, 2001

[2] A. Hyvärinen, E. Oja.
Independent Component Analysis: Algorithms and Applications. Neural Networks, 13(4-5):411-430, 2000

[3] A. Hyvärinen.
Fast ICA for noisy data using Gaussian moments. 1999.

[4] N. Pfister, S. Weichwald, P. Bühlmann, B. Schölkopf.
Robustifying Independent Component Analysis by Adjusting for Group-Wise Stationary Noise. Journal of Machine Learning Research, 1–50, 2019.

[5] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, et al.
A Review of Classification Algorithms for EEG-based Brain-Computer Interfaces: A 10-year Update. Journal of Neural Engineering. IOP Publishing, 15 (3), pp.55. ffhal-01846433f, 2018.

[6] Brunner, Leeb, Putz, Schlögl, Pfurtscheller.
BCI Competition IV. Data sets 2a: <4-class motor imagery> http://www.bbci.de/competition/iv/desc_2a.pdf, 2008.

[7] Tangermann, Müller, Aertsen, Birbaumer, et al.
Review of the BCI competition IV. Frontiers in Neuroscience, https://doi.org/10.3389/fnins.2012.00055, 2012.

[8] Ang, et al.
Filter Bank Common Spatial Pattern Algorithm on BCI Competition IV Datasets 2a and 2b. Frontiers in Neuroscience, 10.3389/fnins.2012.00039, 2012.

[9] P. Comon.
Independent Component Analysis, a new concept? Signal Processing, Elsevier, 36, pp.287-314. ff10.1016/0165-1684(94)90029-9ff. ffhal-00417283f, 1994.

[10] L. Tong, Y. Inouye, R. -w. Liu.
Waveform-preserving blind estimation of multiple independent sources. IEEE Transactions on Signal Processing, vol. 41, no. 7, pp. 2461-2470, doi: 10.1109/78.224254, 1993

[11] T. M. Cover, J.A. Thomas.
Elements of Information Theory. Wiley, 1991.