Cite as: Neuroinformatics Epub ahead of print, December 2012
(or check DOI link for updated information)

Interactions between Pre-Processing and Classification Methods for Event-Related-Potential Classification: Best-Practice Guidelines for Brain-Computer Interfacing

Jason Farquhar and N. Jeremy Hill

Published online: 19 December 2012
DOI: 10.1007/s12021-012-9171-0
PMID: 23250668

Abstract Detecting event related potentials (ERPs) from single trials is critical to the operation of many stimulus-driven brain computer interface (BCI) systems. The low strength of the ERP signal compared to the noise (due to artifacts and BCI irrelevant brain processes) makes this a challenging signal detection problem. Previous work has tended to focus on how best to detect a single ERP type (such as the visual oddball response). However, the underlying ERP detection problem is essentially the same regardless of stimulus modality (e.g. visual or tactile), ERP component (e.g. P300 oddball response, or the error-potential), measurement system or electrode layout. To investigate whether a *single* ERP detection method might work for a wider range of ERP BCIs we compare detection performance over a large corpus of more than 50 ERP BCI datasets whilst systematically varying the electrode montage, spectral filter, spatial filter and classifier training methods. We identify an interesting interaction between spatial whitening and regularised classification which made detection performance independent of the choice of spectral filter low-pass frequency. Our results show that pipeline consisting of spectral filtering, spatial whitening, and regularised classification gives near maximal performance in all cases. Importantly, this pipeline is simple to implement and completely automatic with no expert feature selection or parameter tuning required. Thus, we recommend this combination as a "best-practice" method for ERP detection problems.

Keywords EEG · ERP · BCI · Decoding · LDA · spatial filtering · spectral filtering

Jason Farquhar
Donders Institute for Brain, Cognition and Behaviour
Radboud University Nijmegen, The Netherlands
E-mail: jadref@gmail.com

N. Jeremy Hill
Wadsworth Center, New York State Department of Health
Empire State Plaza, Albany, NY, 12201 USA
E-mail: jezhill@gmail.com

1 Introduction

The aim of Brain Computer Interface (BCI) research (Birbaumer et al, 2000; Wolpaw et al, 2002; van Gerven et al, 2009) is to provide a direct link from human intentions, as observed in brain signals, to control of computers. Such systems might be used to allow completely paralysed people to communicate, among other potential applications. Here we focus on BCI systems for the transmission of an intentional communication or control signal. This requires three challenges to be met: a mental task which evokes or induces some pattern of activation in the brain that can be voluntarily modulated by the user; a way of measuring this activity; and methods for processing the signal to decode the user's intentions as accurately as possible.

Broadly speaking, BCIs can be categorised as either *evoked* or *induced* depending on the type of brain response they exploit. Evoked-response BCIs use the brain's responses to sensory stimuli, modulated by the user's (overt and/or covert) selective attention to those stimuli. The signals that can be exploited in such BCIs are time-locked in that they have a predictable polarity at given times relative to the stimulus event. For example, stimuli may elicit event-related potential components (ERPs) that are attention-modulated, such as the P300, a positive deflection in voltage that occurs 300–400 msec after a target stimulus under certain conditions. This forms the basis of one of the most popular and widely-studied BCIs (Farwell and Donchin, 1988). Steady-state evoked potentials (SSEPs) also exhibit a predictable phase relationship to the oscillatory stimulus signals that generate them, and may also form the basis of such a BCI (Middendorf et al, 2000). By contrast, in *induced*-response BCIs the signal polarity itself is *not* time-locked but some other feature is. Induced responses most commonly use second-order features (Christoforou et al, 2010), such as power or coherence. For example, imagining moving a hand or foot causes a reduction in the power in the μ-frequency range over the appropriate region of the motor cortex. Such a power reduction is called an event-related desynchronisation (ERD) and is the basis for many BCIs that are driven by imagined movement (Pfurtscheller et al, 2006; Pfurtscheller and Neuper, 2001). Other internally-generated mental states, e.g. performing mental arithmetic, also tend to cause frequency- and location-specific changes in signal power. The different single features used in these two types of BCI significantly change the type of signal processing needed to decode the users' intentions.

Detecting the BCI relevant brain signal is a challenging problem due to their small strength relative to the background noise (for the purposes of the current paper we will refer to any non-BCI task related signal as "noise"). Signal processing techniques are needed to pre-process the raw sensor data to suppress the non-BCI signals, such as line-noise or muscle artifacts, whilst keeping the BCI as strong as possible, i.e. to maximise the signal-to-noise ratio. Commonly-used pre-processing methods include: temporal windowing to select time-ranges of interest; spectral-filtering to select frequency ranges where the signal lies; and spatial-filtering to suppress signals from unwanted locations and select electrode locations where the signal of interest should

be strongest (Blankertz et al, 2011). Due to the large degree of between-subject variability in BCI signal properties a subject-dependent classifier is also normally used to detect which task the user was performing, or whether a given stimulus was attended or unattended. The classifier is trained using pre-processed data from a calibration session, in which the user performs a pre-determined sequence of mental tasks or selections.

Getting the pre-processing and classifier training pipeline right is critical to maximising BCI performance. Ideally, one would like a system which could determine the optimal parameter settings automatically for any BCI given only the calibration-session data. However, this is difficult given the large differences in signal properties for different BCIs due to different sensor types and mental task used. For example, evoked BCIs need only first-order features whereas induced BCIs require second- or higher-order features.

In this paper, we focus on the more limited problem: "*What is the best pre-processing and classification pipeline for classification of ERPs measured by non-invasive electrophysiological sensors*?"

Note the two restrictions. Firstly, only ERP classification problems are considered—thus the BCI-relevant signal is assumed to be a first-order feature of the measured sensor data with a prototypical temporal and spatial distribution time-locked to the stimulus presentation. Secondly, only non-invasive electrophysiological sensors are considered, such as electroencephalogram (EEG, a safe, cheap and therefore popular non-invasive method) or magnetoencephalogram (MEG, an expensive and non-portable, but powerful, research tool). These sensors have high temporal resolution, but low spatial resolution since they suffer from spatial blurring where, due to *signal propagation* and *volume conduction* effects, each sensor delivers a mixture of signals from multiple sources (Nunez and Srinivasan, 2005). Importantly, this mixing process is, to a very close approximation, linear with respect to signal intensity and hence can (in principle) be inverted by a linear transformation of the multi-channel array of sensor measurements. An important consequence of these restrictions is that only linear methods are necessary in the pre-processing and classification pipeline. This potentially simplifies the signal-processing pipeline, because (a) the order of temporal and spatial filtering does not matter, and (b) as we show in Sec 2.6, a linear classifier might in principle be able to perform subject-dependent spatial and/or spectral filtering automatically.

1.1 Related work

There have been many previous articles looking at how to best classify ERPs. For example Müller et al (2003), Krusienski et al (2006), Lotte et al (2007) and Selim et al (2008) present comparisons of different classifiers, and Brunner et al (2007) examine different spatial filtering methods. However, few articles look at the interaction between the pre-processing and classifier training method used.

Most similar to our work is Krusienski et al (2008) which investigated the effect of electrode montage, type of reference, and decimation factor on classification performance for visual speller data. Previous work by the same group (Krusienski et al, 2006) recommended step-wise linear discriminant analysis (`swLDA`) (see §2.6) so only this classifier was assessed. For this classifier they investigated the effect of limiting the number of selected features selected on performance. To ensure relevance of the results, all analyses were performed in a session to session context. Krusienski et al (2008) main conclusion was that a combination of an ERP specific 16 channel electrode montage, a decimation factor of 12 (which for their raw sampling rate of 240Hz is roughly equivalent to low-pass filtering below 10Hz and downsampling to 20Hz), and a limit of 60 features gave best performance. Reference choice had no significant effect.

Blankertz et al (2011) has a similar aim to this work – they aim to show how best to classify ERP data. However, they focus on feature selection and comparing classifiers based on Linear Discriminant Analysis `LDA`, with a particular emphasis on shrinkage `LDA` (see §2.6). They also present an analytical method to estimate the optimal regularisation parameter based on Ledoit and Wolf (2004). Blankertz et al (2011) includes a highly informative discussion on how to interpret the solution found by a linear classifier and its relationship to spatial-filters and spatial patterns. In particular they show that to maximise performance the classifier requires knowledge of the interfering noise signals as well as the signal of interest. Thus, it is common for a classifier to give significant weight to electrodes measuring non-class relevant noise sources (such as visual alpha), as well as those measuring the class-relevant signal. Before attempting to interpret the feature weightings learned by a linear classifier one should read this work.

This paper goes beyond these previous works in 2 major ways. Firstly we systematically investigate the effect of changing different pre-processing and classification possibilities on the whole system's performance. Secondly we compare performance on *different types of ERP* gathered with different measurement devices in different labs. Thus, we aim to derive a single "best-practice" pre-processing and classification pipeline which should give near-optimal performance on a wide range of ERP BCIs.

The rest of this paper is arranged as follows: §2.1 describes the various data-sets we have used, the types of pre-processing and classifier training methods we compare (§2.2-§2.6) and the evaluation methodology (§2.7). §3 presents the results and §4 summarises and interprets them. Finally, §5 gives our recommended ERP classification pipeline and finishes with some concluding thoughts.

2 Methods

The aim of this paper is to identify a "best-practice" pre-processing and classification pipeline which gives near-optimal performance on a wide range of ERP classification problems. To do this requires both a representative range

Dataset (reference)	Description	Sensor	#Subj	Raw Size ($d \times T \times N$)
`vgrid` (Hill et al, 2008)	visual speller, target vs. non-target	EEG	14	61×150×1000
`audio` (Hill et al, 2005)	auditory ERP, left vs. right attention	EEG	5	40×153×370
`comp` (Blankertz et al, 2006, II)	visual speller, target vs. non-target	EEG	2	64×144×1020
`epfl` (Hoffmann et al, 2008)	visual speller, target vs. non-target	EEG	8	32×204×1114
`tactile` (Hill and Raths, 2007)	tactile evoked ERP, left vs. right attention	MEG	10	150×375×480

Table 1 Summary of the major properties of the ERP datasets used in this study. Raw datasets contain N example ERPs after balancing each of which has d channels sampled at T time-points.

of ERP problems and a methodology to compare possible pipelines on these problems.

2.1 Datasets used for evaluation

Ideally, we would like to identify a classification pipeline which is generally applicable, and not specific to one particular type of ERP or hardware/software configuration. Thus, we have combined various public and private domain datasets to construct an evaluation set. The key characteristics of these datasets are summarised in Table 1, see the references therein for more detailed information about each experiment. These datasets represent a wide range of possible ERP BCIs with data recorded at different labs, using different hardware and targeting different types of task and stimulus modalities.

Each dataset was prepared for classification by taking a fixed temporal window around each stimulus event which was large enough to capture the class-dependent ERP of interest. To reduce spectral filtering artifacts, an additional buffer of 0.1s was added to both sides of the window and removed after spectral filtering. The windows were also labelled with the class of the event, e.g. left or right attention for the `audio` dataset. The `comp` dataset had many more events than any of the other datasets (6480 events per session for 5 sessions). To limit memory requirements, only data from the first session of each subject was used.

2.2 Pre-processing and classification options considered

To investigate the influence of the pre-processing and classifier selection on ERP classification performance, we conducted a range of off-line simulations where the pre-processing parameters and classifier training methods were sys-

tematically varied. A summary of the basic pipeline used with the various options tested is given here:

1. Montage selection - one of: 8, 16, 32 or all electrodes
2. *DC removal* - Each electrode had its average offset in the window subtracted[1]
3. *Bad electrode removal* - Where a bad electrode is one with more than 3 standard deviations more power than the average electrode power.
4. *Re-referencing* to the common average - by subtracting the average activation over electrodes to reduce the effect of external noise sources, e.g. line noise[2].
5. Spectral filtering - with one of the following low-pass cut-off values: 6Hz, 8Hz, 12Hz, 16Hz, 24Hz or 32Hz
6. Spatial filtering - one of: none, `SLAP`, whitening or ICA
7. Classifier training - one of: `LDA`, `swLDA`, `rLDA` or `rLR`

Stages 2,3 and 4 were fixed for all analyses. For the remaining stages (1,5,6, and 7) a range of options were compared as summarised above. A detailed description of the options considered in these stages is given in the following sections.

2.3 Electrode Montage

Usually before any data is gathered the experimenter selects which sub-set of electrodes to use. The aim here is usually pragmatic, one attempts to trade-off the time required to setup a recording against the likelihood of having at least some of the sensors well placed to detect the signal of interest.

There is, however, more to electrode location selection than positioning them as close as possible to the sources of interest. As discussed in Blankertz et al (2011) a larger number of electrodes, some of them even at appropriate locations to measure noise sources rather than signals-of-interest, allows for more accurate spatial filtering (see §2.5). As against this, a larger number of electrodes also means a larger number of features for classification, which increases the likelihood of overfitting (Duda et al, 2000): we would expect this to affect some classification methods more than others, having a particularly large negative impact on methods with poor complexity control (Duda et al, 2000).

To investigate this effect, 4 different electrode montages were used containing 8, 16, 32, or all available electrodes (see Figure 1). These montages were picked to maximise head coverage for the given electrode count. For a particular ERP generated in a known location it has been shown (see e.g. (Krusienski et al, 2008)) that better performance is attained for a given number of electrodes

[1] This is not strictly necessary as the later spectral filter will also remove the DC, however doing it early improves numerical stability and prevents filter ringing artifacts.

[2] Again this is not strictly necessary as the later spatial filtering will also remove the common activation, however doing it early improves numerical stability and prevents filter ringing artifacts during spectral filtering.

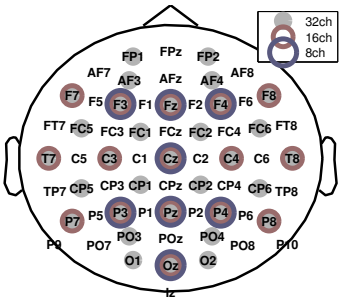

Fig. 1 Illustration of the different montages used to test the effect of reducing the number of sensors on classification performance.

by using ERP-specific montages. However, these generic montages have the advantage that they should perform well across the diversity of different ERP data sets in our corpus.

2.4 Spectral filtering

The aim of any filtering is to improve signal-to-noise ratio by suppressing unwanted noise sources, e.g. line-noise, while leaving the signal of interest intact. Spectral filtering tries to suppress noise based on its frequency characteristics. In theory, given the signal of interest (or its power spectrum) and the power spectrum of the (assumed uncorrelated) noise, an *optimal* signal-to-noise power maximising spectral filter can be found by Wiener filtering (Brown and Hwang, 1997) which weights each frequency by the signal-to-noise ratio in that frequency. In practice however, the exact ERP shape and noise spectrum are unknown so the optimal filter is approximated with a band-pass filter. Given the relatively slow nature of common ERP signals, e.g. N1, P2, P300, N400, MRP, the ERP power is concentrated in the low frequencies, say from 1-12Hz, with little signal outside this range. However, there are significant noise sources above 10Hz, such as line-noise ($\approx$50Hz), visual α-oscillations (10Hz) and sensorimotor oscillations ($\approx$12Hz). Below 0.5Hz slow drift artifacts dominate. Thus, a good rule-of-thumb is to use a band-pass between 0.5 and 10Hz to maximise signal-to-noise-ratios for ERP detection. Indeed, Krusienski et al (2008) found that a sampling rate of 20Hz performed best – which corresponds to a maximum signal frequency of 10Hz.

As well as improving signal-to-noise ratios an additional benefit of low-pass filtering is that after filtering the number of features can be reduced without losing information. The sampling theorem proves that no additional information is gained by sampling a signal more rapidly than twice its maximum frequency (the Nyquist frequency). Thus, a low-pass filtered signal can be down-sampled to 2 times its cut-off frequency without information loss.

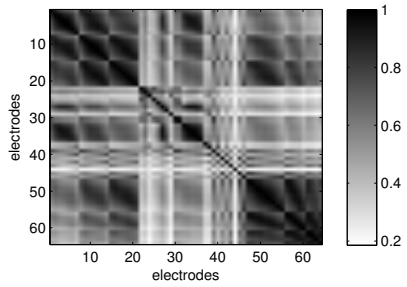

Fig. 2 Covariance between sensors measuring brain-signals using EEG in a 64 electrode montage demonstrating the high correlation between signals measured with different sensors.

Here, we down-sample to 3 times the low-pass cut-off to avoid aliasing of the signal which remains above the 3 dB cut-off frequency.

To investigate the sensitivity the ERP classification performance to the selection of the spectral filter we systematically varied the low-pass boundary from, 6, 8, 12, 16, 24 to 32Hz for a fixed high-pass frequency of 0.5Hz. In all cases the filter is implemented using a Fourier filter where the signal is first Fourier transformed then a weighting is applied to suppress or remove unwanted frequencies and the weighted signal inverse Fourier transformed. After filtering the data was down-sampled to 3x the low-pass cut-off frequency (18, 24, 36, 48, 72 and 96Hz respectively) using Fourier re-sampling, and the 0.1s buffer at the edges of each temporal window (see §2.1) deleted to reduce filtering artifacts.

2.5 Spatial Filtering

As with spectral filtering, the aim of spatial filtering is to improve signal to noise ratios by suppressing noise sources. However, in this case due to volume conduction, one cannot assume that the signal and noise are *spatially* uncorrelated. Indeed, quite the opposite occurs as signal and noise overlap to a large extent in the recorded sensor measurements, as illustrated by the high correlations between sensors in Figure 2.5.

The presence of correlated noise complicates the problem of determining an optimal filter. In particular, correlated noise cannot simply be suppressed by giving the noisy features low weight as in spectral filtering, but must be actively detected and cancelled. As discussed by Blankertz et al (2011), to perform such noise cancellation an optimal spatial filter needs both a large positive weighting where the signal-to-noise is high (i.e. near the signal source) and a large negative weight where the noise-to-signal is high (i.e. near the noise sources) in order to estimate and subtract the noise contribution. Thus to estimate an optimal spatial filter for a signal of interest, one needs to know not only how strongly the signal is detected at each sensor but also how strongly

every interfering noise source is measured at each sensor (and, if there are more sources than sensors, each source's relative strength).

This detailed information about the signal and noise sources is highly subject and environment dependent making it difficult to identify optimal spatial filters in advance of measurements. Further, the optimal spatial filter depends on the spatial pattern of the class-dependent signal to be detected[3].

Thus we have two options. The first option is to use class information to find spatial filters which are optimal for detecting the class-dependent signal of interest. This implicitly requires two supervised training steps - one to determine the spatial filters, and another to train the classifier. Whilst this approach has been very successful in ERD classification where the common spatial patterns (CSP) (Ramoser et al, 2000) method is widely used, in the authors' view training two classifiers is inelegant and increases the risk of *over-fitting*. The second option is to determine the spatial filters in a class-independent manner which somehow makes the later classification problem easier. This approach is more commonly used in ERP classification, where for example ICA is used to identify and cancel eye artifacts.

In this paper we compare 3 unsupervised spatial filtering methods: two methods which reduce spatial correlations; Surface Laplacians and Spatial whitening, and one which further unmixes the sources, Independent Component Analysis—specifically InfoMax ICA (Bell and Sejnowski, 1995). We also include the raw unfiltered data as a control to see if spatial filtering is needed at all. Note: technically, the earlier re-referencing stage means that even the "raw" data has been spatial filtered with the common-average filter.

SLAP One of the significant characteristics of non-invasive electrophysiological measurements of brain activity is that signal propagation means that the signal is blurred considerably relative to that of the cortex below. The scalp surface Laplacian (`SLAP`) is one method to reduce this blurring and increase spatial resolution. The `SLAP` works by detecting peaks in the scalp potential using the sum of its 2nd derivatives. Fundamentally, this is a measure of the local current density flowing radially through the skull into the scalp (Nunez et al, 1994). Hence, it is also called the scalp current density or current source density estimate (CSD). Because current flow through the skull is almost exclusively radial, the `SLAP` has been shown to be a good approximation of the dura potential (Nunez et al, 1994).

To compute the `SLAP` we use an efficient implementation which pre-computes a spatial filter matrix based only on the electrode locations using the spherical spline interpolation method of Perrin et al (1989).

Whitening A consequence of the source mixing process is that nearby electrodes' measurements become highly correlated as they detect mostly the same sources. Removing this correlation will undo part of the mixing process.

[3] The vector of source detection strengths over sensors is called a spatial pattern. Given a matrix of all sources spatial patterns an optimal spatial filter for each source can be computed by taking the pseudo-inverse of this matrix, see (Blankertz et al, 2011).

A whitening (or sphering) transformation removes correlations by linearly re-weighting the sensors so the data is spherically symmetric - such that the transformed "virtual sensors" have unit power and are all uncorrelated.

The whitening spatial filter matrix can be readily computed using the matrix square-root of the sensor covariance matrix. To see this, note that the sensor covariance matrix is given by, $\Sigma_X = XX^\top$, where X is the $[d \times (TN)]$ matrix obtained by concatenating all examples together column-wise. Pre-multiplying X with the inverse matrix square-root of Σ_X and computing the transformed sensor covariance we find,

$$(\Sigma_X^{-1/2}X)(\Sigma_X^{-1/2}X)^\top = \Sigma_X^{-1/2}XX^\top\Sigma_X^{-1/2} = \Sigma_X^{-1/2}\Sigma_X\Sigma_X^{-1/2} = I \qquad (1)$$

where I is the identity matrix. Thus, spatially filtering X with $\Sigma_X^{-1/2}$ maps from raw sensor readings to a new space where the sensors are uncorrelated and have unit power.

Intuitively, one can think of whitening as modifying the data such that all sources have equal power. Thus, if the class-dependent source is very weak relative to the noise sources, e.g. because it is deep inside the skull, the effect of whitening is to increase the relative strength of the class-dependent source, whilst reducing the strength of the higher power noise sources. Conversely, if the class-dependent signal is very strong relative to the noise sources then the effect of the whitening is to reduce the strength of the class-dependent source whilst increasing the strength of the noise sources. In practice, it is likely that both conditions apply, where the class-dependent source is of intermediate strength, with some stronger noise sources and some weaker. Thus, depending on the relative strength of the class dependent source we would expect whitening to help in some cases and hurt in others.

ICA Whilst whitening ensures the transformed sensor measurements are uncorrelated, ICA (as the name implies) ensures the stronger constraint that the transformed measurements are statistically independent (Bell and Sejnowski, 1995; Makeig et al, 1996; Hyvärinen and Oja, 2000). Assuming the "true" sources are independent, this stronger constraint means ICA is more likely to find these true sources. As it enforces a weaker constraint whitening can be seen as a "poor man's ICA" – indeed the first step in most ICA algorithms is to first sphere the data before attempting to find a rotation which maps onto the "true" independent sources (Makeig et al, 1996). To test if this additional computational effort is necessary we compared performance with ICA spatial filtering, specifically as found using the InfoMAX method (Bell and Sejnowski, 1995).

2.6 Classifier training

In formal terms, the ERP classification problem is to determine from a single epoch of recorded brain data if it contains a response which indicates the user

was performing a particular mental task. Let $X \in \mathbb{R}^{[d \times T]}$ be the signal output by the pre-processing pipeline containing d (virtual) sensors and T time points. Further, assume the brain-response can only be one of 2 classes, denoted $+$ and $-$, then the ERP classification problem is to find a function $f(X)$ which maps from X into a predicted class ($+$ or $-$). There are many possible mapping functions, however the simplest is to apply a linear weighting to X and take the sign of the result. That is,

$$\hat{y} = \text{sign}[\sum_{i,j} X(i,j)W(i,j) + b] = \text{sign}[X(:)^\top W(:) + b] \tag{2}$$

where, $W \in \mathbb{R}^{[d \times T]}$ is a weighting matrix, b is a bias term (or threshold), and $Z(:)$ denotes the vectorizing operation which converts a matrix into a column vector by stacking its columns.

W can be thought of as a template for the *difference* between the positive class response and the negative class response[4] and b is a threshold which determines how similar the data X must be to this template for it to match the positive class. Note that W is a weighting over both space and time.

As an aside, notice that all the pre-processing and the classifier application stages are linear operations, that is they work by forming weighted sums of the input data. Thus, in principle, all the classification pipeline stages can be combined into a single weighting matrix W_{all} which transforms raw sensor measurements into classifier predictions in a single step. Conversely, this also implies that the weighting W learnt during classifier training can be though of as implicitly performing additional spatial and spectral filtering of the pre-processed data. This observation, leads to an obvious question: "*As the classifier can learn it, is any of this pre-processing necessary at all?*". To which the answer is, "in-principle, no". In practice, this direct learning approach tends to perform poorly as the signal-to-noise ratio of the data used to train the classifier is higher, leading to over-fitting problems. In essence, this approach requires the classifier to learn what we already know a-priori, e.g. that slow-drifts, line-noise, and high frequencies are all pure noise[5].

However, this does lead to a more relaxed view of the purpose of the pre-processing, which is now not so much to "optimally filter to maximise the signal-to-noise ratio", but more to "transform the data such that the classifier training can find the right weight matrix as easily as possible". Clearly, a signal-to-noise ratio maximising pre-processing achieves this new objective. However, if the classifier training method is good, we should be able to achieve the same performance level with simpler pre-processing and less hand-tuning of parameters. This more relaxed view ties nicely with the aim of this paper in identifying a single ERP classification pipeline which works well for a wide range of ERP problems.

[4] Indeed, a *prototype classifier* uses exactly this method to find W, i.e. $W = \text{mean}(X_+) - \text{mean}(X_-)$.

[5] As a further aside, a second implication of this observation is that by combining operations in this way, the computational cost of applying the classification pipeline on-line can be considerably reduced, to only $d * T$ floating point operations per epoch.

There is a vast literature in the machine-learning community on how to best learn a linear classifier, and many different algorithms have been tried for ERP classification (Krusienski et al, 2006; Lotte et al, 2007), e.g. LDA, swLDA, SVM, NN. Non-linear classification methods have also been tried for ERP classification, but have generally been found to have little or no performance advantage (Müller et al, 2003; Krusienski et al, 2006). Thus we focus on linear methods.

Here 4 methods are compared: two methods, `LDA` and `swLDA`, commonly used for ERP classification, (see for example Krusienski et al, 2006), and two methods, `rLDA`, `rLR`, which include a regularisation term. Regularised classifier training forms the basis of current machine learning research because it has been empirically validated to improve noise robustness by avoiding overfitting in a wide range of problem types (Schölkopf and Smola, 2001). These algorithms are discussed in more detail below.

LDA Linear Discriminant Analysis is also sometimes called Fisher's Discriminant Analysis tries to find a linear transformation of the data which maximises the variance between classes whilst minimising the variance within each class (Duda et al, 2000). The weight matrix for this transformation can be found using

$$w = \Sigma^{-1}(\mu_+ - \mu_-), \tag{3}$$

where, $\Sigma = XX^\top$ is the whole data covariance matrix, μ_+ and μ_- are the means of the positive (resp. negative) class examples. Note, that in this case $X = \mathbb{R}^{[(dT)\times N]}$ is a feature-dimensions by number of examples matrix. Thus each column of X is a complete epoch consisting of all sensors at all sampled time-points.

Under the assumption that the features in each class are generated from Gaussian distributions with the same *known* covariance but different means it can be shown (Duda et al, 2000) that `LDA` is optimal, in that it minimises the misclassification rate. As shown in Blankertz et al (2011) the assumption of a common class covariance with different means is well met for EEG ERP classification problems.

One issue with (3) occurs when the problem is under-constrained because there are more feature dimensions than examples, i.e. the problem is under-sampled. In this case the total data covariance matrix, Σ, is rank-deficient and not invertible. Many methods have been developed to "fix" `LDA` to work in this case including: using only a sub-set of the input features as is done in `swLDA`, using PCA for dimensionality reduction prior to `LDA` in PCA-LDA, adding a so-called *ridge* to the covariance matrix to make it invertible as is done in regularised LDA (`rLDA`), or using the matrix pseudo-inverse. For a review of `LDA` methods on under-sampled problems see Krzanowski et al (1995). In this paper we use `LDA` to denote LDA with the pseudo-inverse fix.

Note that for binary classification problems the `LDA` (and regularised `LDA`) solutions can be found using (regularised) Least squares regression (Duda et al, 2000) by modifying the target labels to be $y_+ = 1./N_+$ and $y_1 = -1./N_-$,

where N_+, N_1 are the number of positive (resp. negative) training examples. Due to its simplicity and computational advantages (one does not need to compute or invert the data covariance matrix) all the `LDA`solutions in this paper are found using this equivalent least squares formulation. The pseudo-inverse LDA solution is equivalent to *min-norm* least squares solution. Pseudo-inverse LDA has also been shown to be equivalent to Uncorrelated LDA (Ye, 2006).

swLDA To cope with under-constrained problems, Stepwise Linear Discriminant Analysis uses only a sub-set of the input features such that Σ is full-rank. Features are selected heuristically in a forward-backward process. Starting from no features, new features are added incrementally if they exceed a threshold p-value (p_{ins}) and removed if they exceed a different higher p-value (p_{rem}). This process is repeated until either a maximum number of features is reached, or a stable set of features found. Note, whilst not a regularisor as the term is used in this paper, this type of *feature selection* is clearly a form of capacity control as it limits the number of features available to the classifier, and hence should also help prevent over-fitting.

`swLDA` was included in this analysis as it is widely used for ERP classification. In part this popularity is based on the comparison of different ERP classification methods in Krusienski et al (2006) which concluded that `swLDA` had the greatest potential. Following the recommendation of Krusienski et al (2006) we use $p_{ins} = 0.1$, $p_{rem} = 1.5$ with a maximum of 60 features for `swLDA`s parameter settings.

rLDA An alternative method to allow LDA to work on under-constrained problems is simply to "fix" Σ to make it invertible by adding a fixed constant "ridge" to its diagonal entries, i.e. $\hat{\Sigma}_r(\lambda) = \Sigma + \lambda I$, where λ is the ridge magnitude. An alternative formulation of `rLDA` discussed in (Blankertz et al, 2011) is *shrinkage*-`LDA` where $\hat{\Sigma}_s(\gamma) = (1-\gamma)\Sigma + \gamma\nu I$ and ν is the average feature variance. `rLDA` and shrinkage-`LDA` are equivalent upto scaling, i.e. $\hat{\Sigma}_s(\gamma) = (1-\gamma)\hat{\Sigma}_r(\nu\gamma/(1-\gamma))$.

As well as making the covariance matrix invertible, the ridge has the effect of regularising the classifier to use only high variance (or high power) signals in the data. To see why, consider the case of uncorrelated features. In this case Σ is a diagonal matrix with the variance of each feature along the diagonal, i.e. $\Sigma = \mathtt{diag}(\mathbf{v})$, with inverse $\Sigma^{-1} = \mathtt{diag}(1/\mathbf{v})$. Thus, the inverse magnifies features with low variance and reduces features with high power. The addition of the ridge changes the inverse to be $\mathtt{diag}(1/(v+\lambda))$ which leaves the high power features relatively unaffected but dramatically reduces the magnification of low power features. Correlated features can be treated in a similar way by un-correlating (but not sphering) them first. Thus, by reducing the magnification of low-power features the effect of adding the ridge is to bias the classifier to prefer high-power features.

Another interpretation of the effect of the ridge on the `LDA` solution is given in (Blankertz et al, 2011) where it is shown that the ridge interpolates between a *univariate* solution ($\gamma = 1$) where each feature is treated independently, and a

multivariate solution ($\gamma = 0$) which takes account of the empirically estimated correlations between features.

rLR As mentioned above `LDA` is provably optimal under the assumption of per-class Gaussian distributed features with a common covariance. If these assumptions are violated then `LDA` may perform poorly, and a classifier which makes fewer assumptions may be appropriate. The statistics and machine learning literature contain many more general classifier training methods, such as quadratic discriminant analysis (QDA) which generalised `LDA` to allow for non-equal class covariances. Many of these more general methods have also been tried on ERP classification problems with mixed results, see for example (Müller et al, 2003; Lotte et al, 2007).

Regularised `LDA` is a *generative* classifier. This means that it models the joint distribution of the data, $\Pr(y, \mathbf{x})$, where y is the class label and $\mathbf{x}$ are the data features. *Discriminative* classifiers by comparsion model only the conditional distribution of the class labels given the features, $\Pr(y|\mathbf{x})$. As they make fewer modelling assumptions, discriminative methods are more robust to model mis-specification – only the model of $\Pr(y|\mathbf{x})$ need be correct. However, it has been shown (Ng and Jordan, 2002; Bouchard and Triggs, 2004) that this robustness may only be useful for large training-set sizes as (if the generative model is approximately correct) generative models have lower classification error on small training sets.

Here we use Logistic Regression (LR) (Duda et al, 2000) as a representative discriminative classifier because it is the natural discriminative counterpart of `LDA` (Ng and Jordan, 2002) and has previously been shown effective for BCI classification problems (Tomioka et al, 2007). Specifically, we use regularised LR (`rLR`), which is a variant of the Logistic Regression to which an additional quadratic regularisation term has been added. Such a regularisor can be seen as either imposing a 0-mean, λ variance spherical Gaussian prior over the classifier weights, or as a weighted quadratic penalty term on the classifier weights. In either case, as for `rLDA`, the net effect of the regularisor is to bias the classifier to prefer high power features.

2.7 Evaluation Methodology

To assess the importance of different combinations of pre-processing and classification we adopted a simple brute-force approach - that is every possible combination of pre-processing options was run on every dataset and the final classification performance evaluated.

A pipeline's performance was estimated using a temporal split-half cross-validation. In this approach the dataset is split into 2 parts (representing the first and second parts of the experiment) and then one part is used to optimise pipeline parameters (i.e. train classifiers, estimate spatial filters) and the other used to test the trained pipeline. The average of the test-set performance over the two halves gives the estimated pipeline performance. Temporal split-half

analysis was used as it maximises the temporal distance between the training and testing sets, and hence is most representative of an actual BCI usage scenario. When necessary, hyper-parameters (such as classifier regularisation parameters) were optimised using a nested 5-fold cross-validation within the training data.

The visual speller datasets are very unbalanced, with 5 times more non-target events than target events. Therefore, to speed classifier training and make the classification performances comparable between datasets, the classes were artificially balanced by randomly discarding examples from the larger class until each class had equal numbers of examples.

2.8 Training set size

Performance of any learning system depends critically on the number of examples used to tune its parameters. However, generating training examples adds an overhead for using the system which is frustrating for the user and reduces the time it can be used productively. Thus minimizing the number of training examples required to achieve acceptable performance is critically important to practical BCI performance.

To investigate this effect, we used 5 different training set sizes containing 50, 100, 200, 400, or all available training examples. To best simulate an experiment with only this number of examples, these reduced training sets were obtained by starting from the normal split-half training set and then discarding all but the first N examples. If the split-half training set was smaller than the desired size then the entire split-half training set was used.

3 Results

Evaluating all possible pipelines on all datasets gives almost 15,000 performance estimates. Thus, we present only our major findings from analysing this performance data.

3.1 Default pipeline

Before examining the effect of changing pre-processing parameters on performance we start by looking at performance using a default classification pipeline. This default pipeline serves two purposes: firstly it acts as a control condition to illustrate the effect of changing the different pipeline options on performance, and secondly it provides a set of default parameters for pipeline stages not explicitly manipulated in the later analyses.

Based on the suggested settings in Krusienski et al (2006) the parameters used in the default pipeline were:

1. Electrode montage: all channels

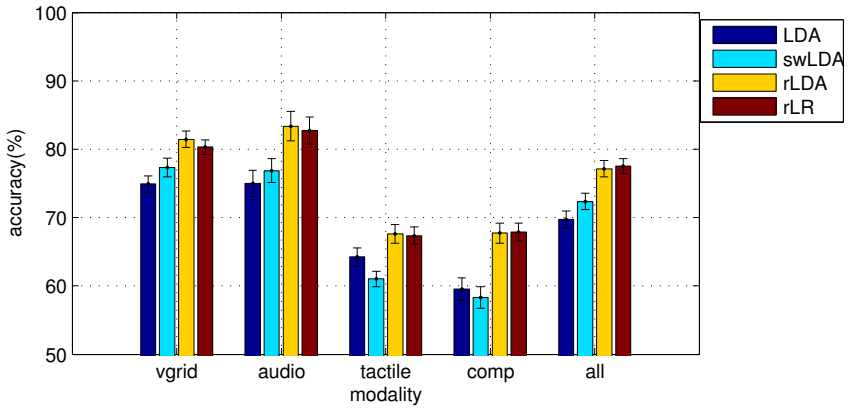

Fig. 3 Average classification accuracy over subjects using the default pipeline for each of the different datasets described in §2.1 and the average over all datasets for the different classifiers. Error-bars indicate standard error over subjects.

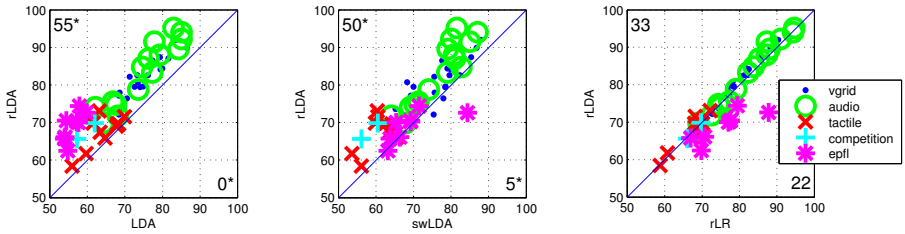

Fig. 4 Performance comparison of `rLDA` against `LDA`, `swLDA` and `rLR` using the default processing pipeline. Each point represents a subject from the indicated dataset. Points above the line indicate better performance for `rLDA`. Numbers in the top-left and bottom-right of each plot represent the number of times the vertical (resp. horizontal) axes method performs better than the other axes. $*$ indicates a significant difference ($p < .05$) using a two-tailed binomial test.

2. DC removal
3. Bad electrode removal
4. Re-referencing to the common average
5. Spectral filtering - with a 0.5–12Hz pass-band followed by down-sampling to 36Hz
6. Spatial filtering - none
7. Classifier training - one of: `LDA`, `swLDA`, `rLDA` or `rLR`

The performance of the 4 classifiers tested with this default pipeline is presented in Figure 3. As these results show, the performance varies widely over the different datasets, ranging from ≈60% correct in the `tactile` and `comp` datasets to more the 80% correct for the `audio` and `vgrid` datasets. Comparing the different classifiers, we see that the regularised classifiers `rLDA` and `rLR` consistently out-perform the un-regularised classifier `LDA`, with the partially regularised `swLDA` performing somewhere in between. The performance advan-

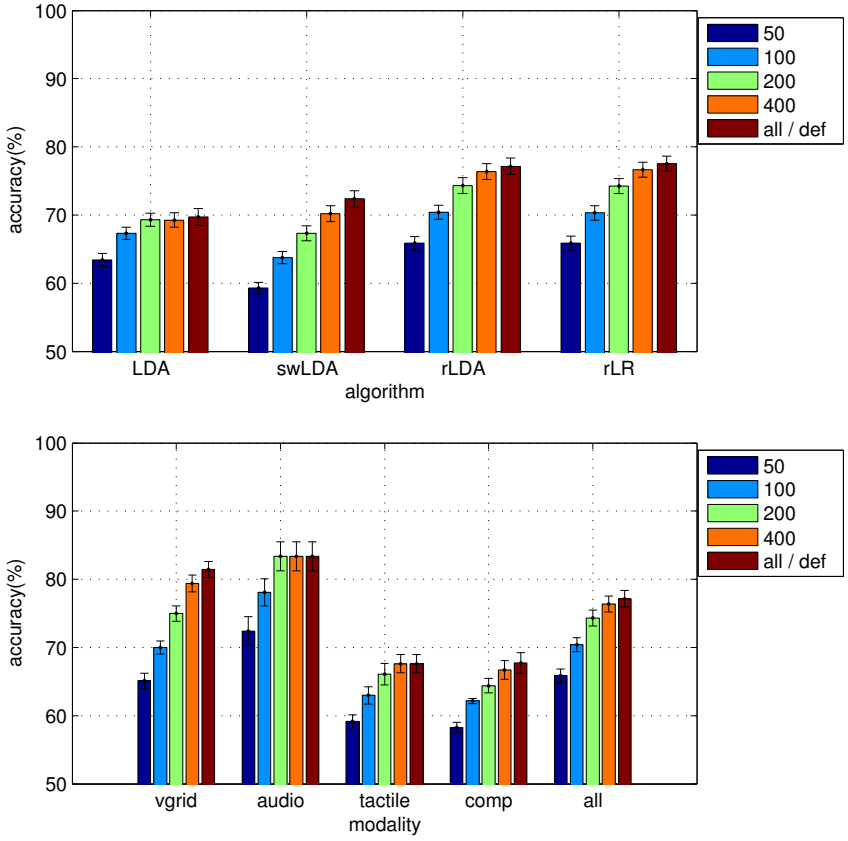

Fig. 5 (Top) The effect varying the number of examples used to train the classifiers on the classification accuracy using different classifiers. (Bottom) The effect of varying the number of training examples on `rLDA` performance broken down by dataset. Error-bars indicate standard errors over subjects. `def` indicates the setting from the default pipeline.

tage for regularised classifiers is highly statistically significant ($p < .001$). In fact as shown in Figure 4 `rLDA` outperforms `LDA` in *all problems* by an average of 10% and better then `swLDA` in *all but 5 problems*. The difference between `rLDA` and `rLR` was not statistically significant. However, Figure 4 shows that `rLDA` performs slightly better than `rLR` is 4 out of 5 of the datasets, the exception being `epfl` where `rLR` is almost 8% better.

3.2 Training set size

The effect of varying the number of examples used to train the classification pipeline is illustrated in Figure 5. Again this shows the general performance advantage of the regularised classifiers which have superior performance for *all* training set sizes. As one would expect, performance improves continuously

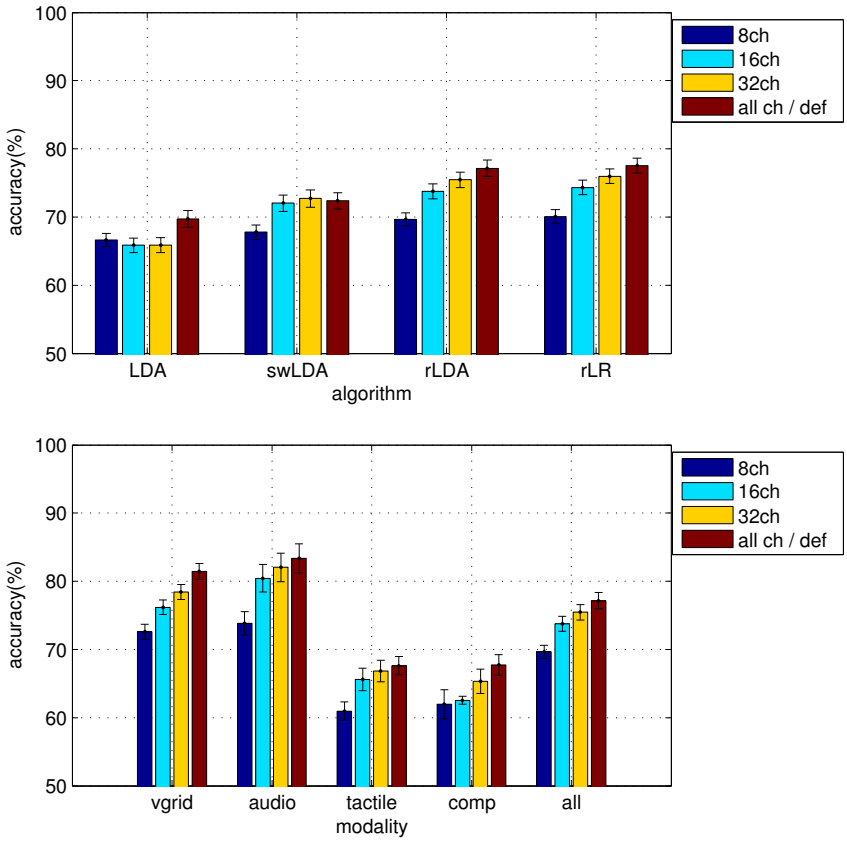

Fig. 6 (Top) The effect varying the number of electrodes used on the classification accuracy using different classifiers. (Bottom) The effect of varying the number of electrodes on `rLDA` performance for each dataset. Error-bars indicate standard errors over subjects. `def` indicates the setting from the default pipeline.

with increasing training set size. In fact performance improves log-linearly, with performance improving by $\approx 4\%$ for each doubling of the training set size - this is most clear in the `vgrid` and `comp` datasets which both have >800 examples allowing all training set sizes up to 400 examples to be tested without saturation.

3.3 Electrode Montage

The effect on classification performance of varying the number of used electrodes is illustrated in Figure 6. Again this clearly shows the distinction between the regularised classifiers, `rLDA` and `rLR`, for which performance only increases with increasing numbers of electrodes, and the un-regularised method (`LDA`) for which performance initially *decreases* with more electrodes but gets

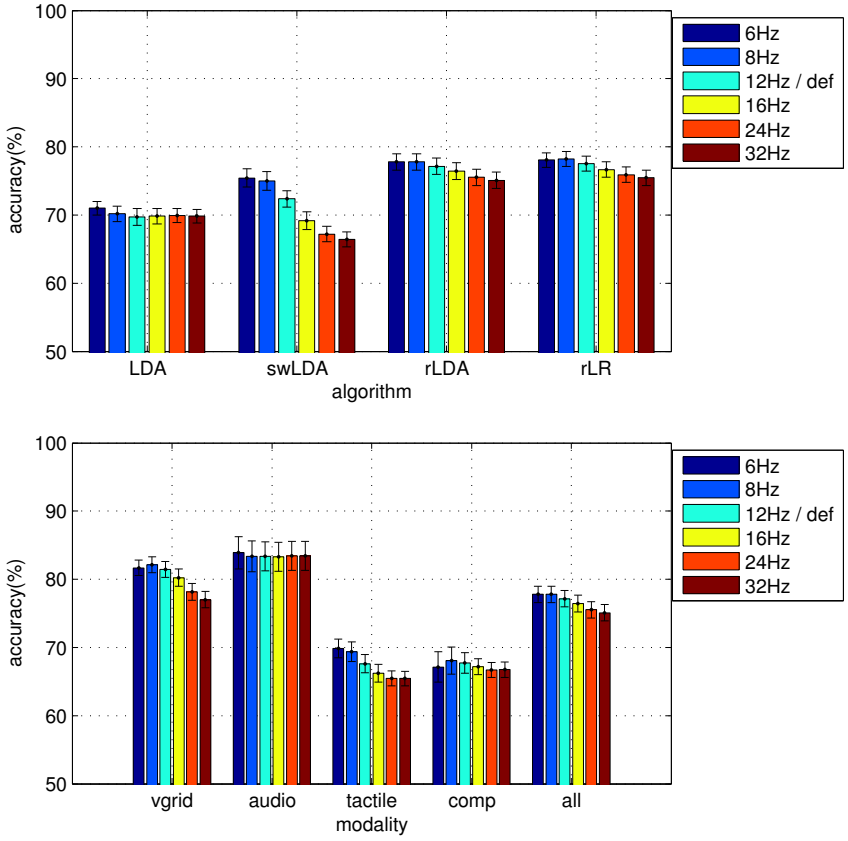

Fig. 7 (Top) The effect of varying the low-pass threshold on classification accuracy using different classifiers. (Bottom) The effect of varying the low-pass threshold for each dataset for the `rLDA` classifier. Error-bars indicate standard errors over subjects. `def` indicates the setting from the default pipeline.

slightly better for the full montage. Again `swLDA` lies between these two extremes with performance getting better from 8 to 16 electrodes, but then remaining static for more electrodes. Looking at the dataset dependence in Figure 6 we see that adding more channels improves performance in all cases. However, this improvement is highly variable across datasets, with the largest improvements of almost 10% `vgrid` and `audio` and the smallest (about 2%) for `epfl`.

3.4 Spectral Filter

The effect of varying the low-pass cut-off frequency on classification performance using the default pipeline is shown in Figure 7. Note: after filtering the data were down-sampled to three times the threshold, thus the number of

features used for classification is proportional to the low-pass frequency. Most striking in these results is that for all methods the performance *decreases* with an increasing low-pass cut-off and number of features. This effect is worst for `swLDA` whose performance decreases by almost 10%, followed by the regularised classifiers which lose about 2% and least for `LDA` which is only $\approx 1\%$ worse for the 32Hz cut-off.

Looking at the per-dataset results, we see that the improvement from having a small low-pass threshold is largest for the `vgrid` and `tactile` datasets. Conversely, the `epfl` and `audio` show almost no effect from varying the low-pass cutoff. Interestingly, these datasets also have the fewest sensors channels (32 and 40 respectively), indicating that getting the right low-pass cut is most important when the input feature dimensionality is high.

3.5 Spatial Filter

The effect of varying the spatial-filtering method on classification performance using the default pipeline is shown in Figure 8. As was the case for varying the montage, these results show a clear distinction between the regularised methods, `rLDA`, `rLR`, where performance *improves* (slightly) when using spatial filtering and the other two methods (`LDA`, `swLDA`) where performance *decreases*. One further sees that for the regularised classifiers spatial whitening and ICA give the same performance and that both outperform `SLAP`. As shown in Figure 9 the performance advantage from spatial whitening over no spatial filtering is very significant ($p < 0.001$). Further, there was no significant difference between whitening and ICA ($p > 0.1$). Indeed, in many cases they performed identically. Looking at the per-subject effects Figure 9 also shows that the only cases in which whitening is a serious disadvantage are two of the `tactile` subjects. Interestingly, `tactile` is also the dataset with by far the largest number of sensors (151).

3.6 Interaction effects: combined spatial and spectral filtering

So far only single pre-processing factors have been considered. However, there are likely to be interactions between different pre-processing stages which influence final performance. Exploration of the results showed two main effects which are illustrated in Figure 10. Figure 10(top) shows that the performance advantage when using spatial whitening and a regularised classifier increases with increasing numbers of electrodes (at least up to 32 electrodes). Figure 10(bottom) shows that when using spatial whitening and a regularised classifier performance no longer deteriorates as the low-pass cut-off frequency is increased.

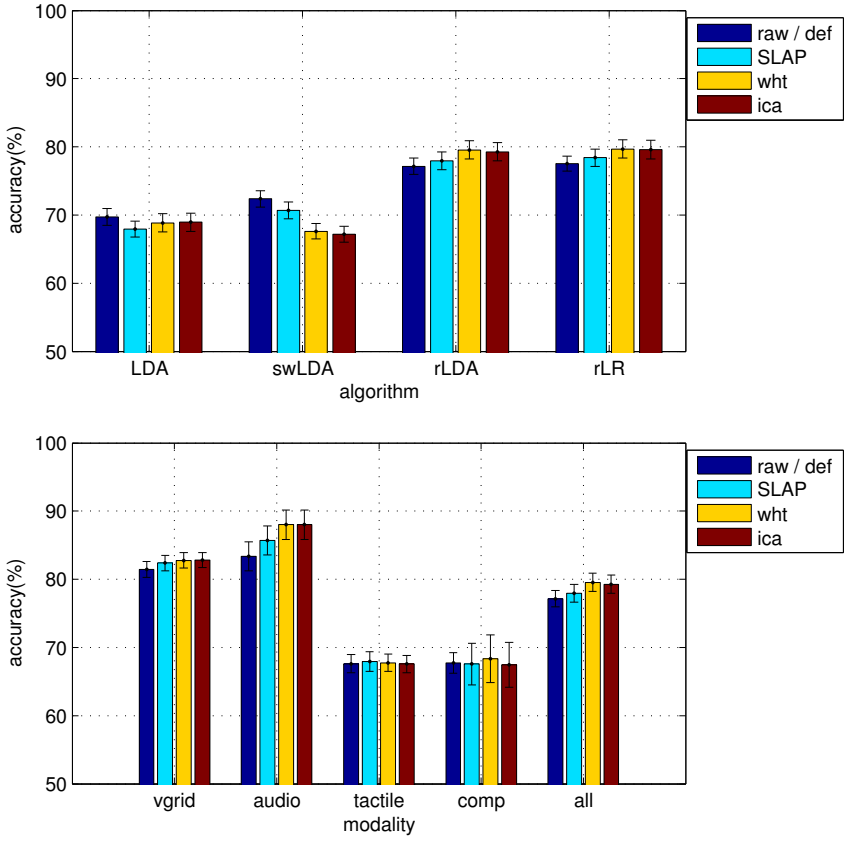

Fig. 8 (Top) The effect varying the spatial filtering method on classification accuracy using different classifiers. (Bottom) The effect of varying the spatial filtering method for each dataset on the `rLDA` classifier. Error-bars indicate standard errors over subjects. `def` indicates the setting from the default pipeline.

4 Discussion

To summarise the above results, we have shown that

1. regularised classifiers perform better than unregularised classifiers in (almost) every dataset when using a reasonable set of default pre-processing parameters.
2. a robust regularised classifier is able to cope with a large number of input electrodes, and in fact *improves* with more inputs. This is in stark contrast to the unregularised `LDA` and `swLDA` classifiers whose performance remains unchanged or even *decreases* with increasing numbers of electrodes.
3. a regularised classifier is robust to mis-specification of the spectral filter parameters and down-sampling rate. Performance can still be improved by using the correct spectral filter, but the penalty from using too high

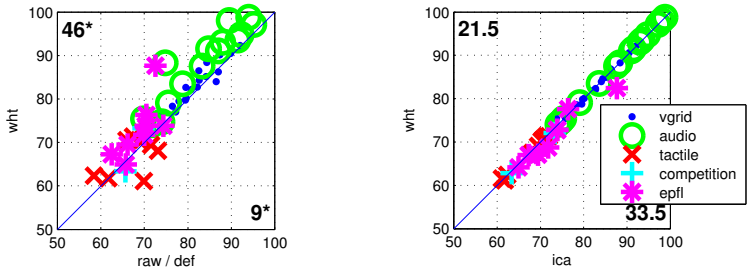

Fig. 9 Performance comparison of spatial whitening (wht) against the `default` pipeline with no spatial filtering (raw) and Independent Components Analysis (ica) when using `rLDA` for classifier training. Each point represents a subject from the indicated dataset. Points above the line indicate better performance for whitening. Numbers in the top-left and bottom-right of each plot represent the number of times the vertical (resp. horizontal) axes method performs better than the other axes (with .5 for an exact draw). * indicates a significant difference ($p < .05$) using a two-tailed binomial test.

a low-pass cut-off is minimal. Again this is in contrast to `swLDA` whose performance degrades rapidly with increasing the spectral filter cut-off frequency.

4. `rLDA` performed slightly better than `rLR` in 4 datasets and significantly worse in 1 dataset.
5. spatial filtering improves classification performance if a regularised classifier is used. Spatial whitening gave the best performance, which was identical to that of ICA.
6. the benefit of spatial whitening increased with increasing numbers of electrodes.
7. if spatial whitening and a regularised classifier are used then performance is *independent* of the spectral filter low-pass cut-off.

The first three of these results will not be surprising to anyone versed in modern statistical learning theory. There is a vast literature of theoretical and empirical research which shows how using regularisation makes classifier training robust to the *over-fitting* caused by increasing the number of input features or noise level. Thus a machine learning researcher would expect regularised classifiers to perform better in the majority of cases, which is exactly what was found in this paper. These results agree with those of Blankertz et al (2011) who also found `rLDA` performed best, but contradict those of Krusienski et al (2008) who found that `swLDA` and `LDA` performed best. This difference can be explained with reference to Figures 6 and 7. These show that as the number of input features is reduced, either by reducing the number for electrodes or downsampling more heavily, the disadvantage of `swLDA` and `LDA` is reduced. Thus, for the configuration considered in (Krusienski et al, 2008) of 8 electrodes and a 20Hz sampling rate one would expect all methods to achieve equivalent performance. Note: the final recommendation for `swLDA` in this paper was based on its ability to work well with large feature spaces through feature selection.

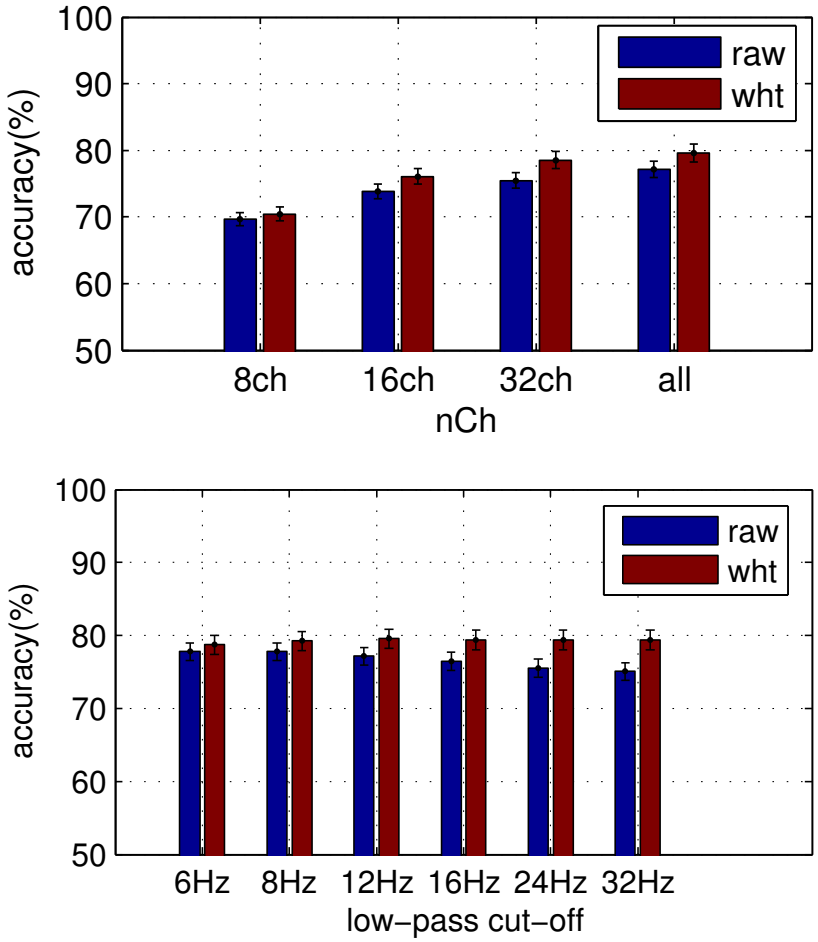

Fig. 10 Interaction effects between pre-processing methods for a `rLDA` classifier. (Top) shows the effect of varying both the number of electrodes in the montage and the type of spatial filter used, (Bottom) shows the effect of varying the spectral filter and type of spatial filter. Error-bars indicate standard errors over subjects.

However, the work presented here and by Blankertz et al (2011) show that `rLDA` is a better solution for such under-constrained problems.

As explained in §2.6 the better performance of `rLDA` compared to `rLR` in most of the datasets indicates that `LDA`s assumption of equal covariance Gaussian distributions is *mostly* correct. Krusienski et al (2008) found a similar result for visual speller ERPs where the discriminative linear SVM performed worse than the generative `LDA` or `swLDA` classifiers. Unfortunately, the anomalous result for `epfl` shows this is not always the case – interestingly, `epfl` is also the only dataset with one session in each of the split-halves.

The results for spatial filtering are more surprising. As described in §2.5 the source-equalisation property of whitening means that whilst it reduces the power of the the few strongest noise sources it also increases the power of the weak noise sources. As there are generally many more weak sources than strong ones, the net effect is to *increase* the total noise strength. Thus one would expect whitening to decrease performance, however our results show exactly the reverse effect, where *if you use a regularised classifier* performance actually improves. Intuitively, this contradiction can be understood by noting that regularised classifiers are more sensitive to high-power noise sources than weak ones – the whole purpose of the regularisor is to suppress low-power sources. Thus for these classifiers the reduction in high-power noise sources is more important than the increase in the low-power sources, resulting in an overall performance improvement. Clearly, there is a potential issue here if the true signal is very strong, or there are very many noise sources. Then the benefit from reducing a few strong noise sources is not enough to overcome the loss from reducing the true-signal and increasing the weak noise sources. Indeed, the performance reductions when using whitening in the `tactile` dataset are probably due to the large number of sensors (and hence noise sources) in this dataset. Fortunately, most BCI problems have weak true signals and few sensors so spatial whitening normally improves performance.

Given the benefits of whitening one might postulate that the superior source separation obtained by ICA should further improve performance. Surprisingly, the results in Figure 9 showed that, despite the significant additional computation effort expended by ICA to find the underlying independent sources, the classification performance was essentially identical. This result can be understood by noting that (as discussed in §2.5) all ICAs can be though of as first spatially whitening the data such that all the new "virtual sensors" are uncorrelated, and then spatially *rotating* the "virtual sensors" with an orthogonal matrix to impose the additional independence requirement (Hyvärinen and Oja, 2000). It can be shown that the predictions of a quadratically regularized linear classifier are invariant to rotation of the input features (see Appendix A). This is because rotating the input features causes the solution weights to rotate in the opposite direction such that both rotations cancel leaving the classifier predictions unchanged (this is similar to how shuffling the order of features shuffles the order of the solution weights). An alternative, more intuitive view of this result is that as (by definition) a rotation cannot change the *strength* of the signal and noise sources, it has no effect on the data signal-to-noise ratio and hence on classification performance. Similar identical performance between ICA and whitening has also been found for object recognition tasks in (Vicente et al, 2007).

Note this invariance to rotations (and hence identical performance between ICA and other whitening transforms) is *only* true when the spatial-filtering rotation is applied directly to the classifier features. This is usual in the context of ERP classification studied in the current paper. For bandpower classification, for example, where the non-linear step of bandpower estimation comes between spatial filtering and classification, this equivalence no-longer holds

and different spatial filtering methods (PCA whitening alone, ICA of various kinds, CSP) may be expected to produce different results, as is familiar from the BCI literature, see for example (Blankertz et al, 2008).

The result that spatial whitening makes (regularised classifier) performance invariant to the choice of spectral filter is perhaps the most surprising result in this paper. To explain this result, first note that due to the inverse frequency ($1/f$) spectrum typical of brain-data, higher frequencies have lower power. Further, note that regularisation biases the classifier towards stronger signals and suppresses low power signals. In non-whitened data this low-power signal suppression suppresses both low-power spatial and spectral sources. Spatial whitening, however, equalises spatial powers so regularisation effects only spectral powers. Thus, after spatially whitening the data by tuning the regularisation strength the classifier is implicitly determining the optimal low-pass filter.

5 Conclusions

The stated goal of this paper was to identify a "best-practice" ERP classification pipeline, which could be used on almost any ERP problem to get near optimal performance. Based, on the results this pipeline is:

1. Use as many electrodes as you can practically record and "let the machine decide" if they are useful.
2. Spectrally filter to remove obvious noise components. A pass band of 0.5-12Hz seems near optimal.
3. Spatially whiten to equalise source powers.
4. Use a linear quadratically regularised classifier training method, such `rLDA`.

This pipeline gave near optimal performance for more than 90% of the datasets we considered and was within a few percent in the other cases.

To save computational resources the the spectral filtering can be followed by down-sampling to 36Hz. Further, as long as the classifier training method is regularised the exact method used has little impact. We recommend `rLDA` as gave the best average performance. `rLDA` is also computationally efficient, is easy to implement using regularised least squares regression (see §2.6), and a good analytic estimate for the regularisation strength can be found using shrinkage LDA (Blankertz et al, 2011).

There are a number of areas where this study could be extended. Firstly, as most current BCIs are re-trained at the start of every session this work only looked at single-subject single-session performance[6]. However, one would like to transfer classifiers between sessions and subjects to minimise or remove the need for this re-training. Further, we only considered simple quadratic regularisation methods here. More advanced classifiers/regularisors could further improve classification performance, by for example automatically determining

[6] This limitation was also due in-part to the sparsity of publicly available multi-session datasets.

the optimal electrode montage (Zou and Hastie, 2005; Meier et al, 2008), or finding a low-rank decomposition of the weight-matrix to reduce the number of parameters to be estimated (Farquhar, 2009; Christoforou et al, 2010). Finally, intuitively the benefit of whitening seems to depend on the strength of the true-signal. Getting a more theoretical understanding of when whitening helps and developing a method to automatically decide if it should be used would increase the general applicability of the "best-practice" pipeline.

Information sharing statement

This work utilised 5 BCI datasets. Two of these datasets are publicly available:

- `comp` is available from the BCI competition III website as dataset II (see `http://www.bbci.de/competition/iii/`)
- `epfl` is available from the EPFL website as the BCI: P300 dataset (see `http://mmspg.epfl.ch/downloads`)

All software used in this paper was written in MATLAB by the authors and is available on request.

Acknowledgements The authors acknowledge the support of the BrainGain Smart Mix Programme of the Netherlands Ministry of Economic Affairs and the Netherlands Ministry of Education, Culture and Science.

A Feature rotation invariance of quadratically regularised linear classifiers.

A quadratically regularised linear classifier finds its solution by minimising an objective function with the form[7],

$$J(w) = \lambda w^\top w + \sum_{i=1..N} \mathcal{L}(x_i^\top w, y_i) \tag{4}$$

where, $x_i \in \mathbb{R}^d$ is the ith training example with d features. $w \in \mathbb{R}^d$ are linear classifier weights. $\mathcal{L}$ is the classification loss function, which penalises differences between the classifier predictions ($x_i^\top w$) and the examples true class y_i. Depending on the choice of loss function $\mathcal{L}$ one obtains different classifiers, e.g. for logistic regression $\mathcal{L} = (1+\exp(-y_i x_i^\top w))^{-1}$, or a least squares classifier $\mathcal{L} = (y_i - x_i^\top w)^2$ (which can be used to implement `LDA` §2.6). $w^\top w$ is the quadratic regularisation penalty, which penalises "complex" solutions and λ the relative strength of this penalty.

Taking derivatives with respect to w and setting equal to zero one finds the optimal solution, w^*, is given by:

$$2\lambda w^* + \sum_{i=1..N} \mathcal{L}'(x_i^\top w^*, y_i) x_i = 0, \tag{5}$$

where $\mathcal{L}'$ is the derivative of loss function $\mathcal{L}$.

[7] We neglect the constant bias-term for simplicity.

If one *rotates* the features with an arbitrary rotation matrix R such that $\hat{x} = Rx$ the solution, $\hat{w}^*$, to this rotated problem is given by:

$$2\lambda\hat{w}^* + \sum_{i=1..N} \mathcal{L}'(\hat{x}_i^\top \hat{w}^*, y_i)\hat{x}_i = 0, \tag{6}$$

$$2\lambda\hat{w}^* + \sum_{i=1..N} \mathcal{L}'(x_i^\top R^\top \hat{w}^*, y_i)Rx_i = 0, \tag{7}$$

$$2\lambda R^\top\hat{w}^* + \sum_{i=1..N} \mathcal{L}'(x_i^\top R^\top \hat{w}^*, y_i)x_i = 0, \tag{8}$$

where we have used the property that the inverse of a rotation is its transpose, i.e. $R^\top R = I$. Making the substitution $R^\top\hat{w}^* = w^*$ one sees that (5) and (8) are identical with the same solution, demonstrating that the only effect of rotation of the features is to rotate the optimal solution in the opposite direction.

References

Bell AJ, Sejnowski TJ (1995) An information-maximization approach to blind separation and blind deconvolution. Neural Computation 7(6):1129–1159, DOI 10.1162/neco.1995.7.6.1129, URL `http://dx.doi.org/10.1162/neco.1995.7.6.1129`

Birbaumer N, Kübler A, Ghanayim N, Hinterberger T, Perelmouter J, Kaiser J, Iversen I, Kotchoubey B, Neumann N, Flor H (2000) The thought translation device (TTD) for completely paralyzed patients. IEEE Transactions on Rehabilitation Engineering 8(2):190–193, DOI 10.1109/86.847812

Blankertz B, Müller K, Krusienski DJ, Schalk G, Wolpaw JR, Schlögl A, Pfurtscheller G, Millán J, Schroder M, Birbaumer N (2006) The BCI competition III: validating alternative approaches to actual BCI problems. IEEE Transactions on Neural Systems and Rehabilitation Engineering 14(2):153–159, DOI 10.1109/TNSRE.2006.875642

Blankertz B, Tomioka R, Lemm S, Kawanabe M, Müller KR (2008) Optimizing spatial filters for robust EEG single-trial analysis. IEEE Signal Processing Magazine 25(1):41–56, DOI 10.1109/MSP.2008.4408441

Blankertz B, Lemm S, Treder M, Haufe S, Müller KR (2011) Single-trial analysis and classification of ERP components : A tutorial. NeuroImage 56(2):814–825, DOI 10.1016/j.neuroimage.2010.06.048

Bouchard G, Triggs B (2004) The tradeoff between generative and discriminative classifiers. In: 16th IASC International Symposium on Computational Statistics (COMPSTAT '04), Prague, Tcheque, Republique, p 721728

Brown RG, Hwang PYC (1997) Introduction to Random Signals and Applied Kalman Filtering, Wiley, TK5102.5.B696, vol 2. John Wiley & Sons

Brunner C, Naeem M, Leeb R, Graimann B, Pfurtscheller G (2007) Spatial filtering and selection of optimized components in four class motor imagery EEG data using independent components analysis. Pattern Recognition Letters 28(8):957–964, DOI 10.1016/j.patrec.2007.01.002

Christoforou C, Haralick R, Sajda P, Parra LC (2010) Second-Order bilinear discriminant analysis. Journal of Machine Learning Research 11:665–685

Duda RO, Hart PE, Stork DG (2000) Pattern Classification, 2nd edn. Wiley-Interscience

Farquhar J (2009) A linear feature space for simultaneous learning of spatio-spectral filters in BCI. Neural Networks 22(9):1278–1285, DOI 10.1016/j.neunet.2009.06.035

Farwell LA, Donchin E (1988) Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. Electroencephalography and Clinical Neurophysiology 70(6):510–523

van Gerven M, Farquhar J, Schaefer R, Vlek R, Geuze J, Nijholt A, Ramsey N, Haselager P, Vuurpijl L, Gielen S, Desain P (2009) The braincomputer interface cycle. Journal of Neural Engineering 6(4):041,001, DOI 10.1088/1741-2560/6/4/041001

Hill J, Farquhar J, Martens S, Bießmann F, Schölkopf B (2008) Effects of stimulus type and of Error-Correcting code design on BCI speller performance. In: Advances in neural information processing systems 21 : 22nd Annual Conference on Neural Information Processing Systems 2008, Corran, Vancouver, BC, pp 665–672

Hill NJ, Raths C (2007) New BCI approaches: Selective attention to auditory and tactile stimulus streams. In: PASCAL Workshop on Methods of Data Analysis in Computational Neuroscience and Brain Computer Interfaces, Fraunhofer FIRST, Berlin

Hill NJ, Lal TN, Bierig K, Birbaumer N, Schölkopf B (2005) An auditory paradigm for Brain-Computer interfaces. In: Saul LK, Weiss Y, Bottou L (eds) Advances in Neural Information Processing Systems 17, MIT Press, Cambridge, MA, pp 569–576

Hoffmann U, Vesin J, Ebrahimi T, Diserens K (2008) An efficient p300-based brain-computer interface for disabled subjects. Journal of Neuroscience Methods 167(1):115–125, DOI 10.1016/j.jneumeth.2007.03.005

Hyvärinen A, Oja E (2000) Independent component analysis: algorithms and applications. Neural Networks 13(4-5):411–430, DOI 10.1016/S0893-6080(00)00026-5

Krusienski D, Sellers E, McFarland D, Vaughan T, Wolpaw J (2008) Toward enhanced p300 speller performance. Journal of Neuroscience Methods 167(1):15–21, DOI 10.1016/j.jneumeth.2007.07.017

Krusienski DJ, Sellers EW, Cabestaing F, Bayoudh S, McFarland DJ, Vaughan TM, Wolpaw JR (2006) A comparison of classification techniques for the p300 speller. Journal of Neural Engineering 3:299–305, DOI 10.1088/1741-2560/3/4/007

Krzanowski WJ, Jonathan P, McCarthy WV, Thomas MR (1995) Discriminant analysis with singular covariance matrices: Methods and applications to spectroscopic data. Journal of the Royal Statistical Society Series C (Applied Statistics) 44(1):101–115, DOI 10.2307/2986198

Ledoit O, Wolf M (2004) A well-conditioned estimator for large-dimensional covariance matrices. Journal of Multivariate Analysis 88(2):365–411, DOI 10.1016/S0047-259X(03)00096-4

Lotte F, Congedo M, Lcuyer A, Lamarche F, Arnaldi B (2007) A review of classification algorithms for EEG-based brain-computer interfaces. Journal of Neural Engineering 4(2):R1–R13, DOI 10.1088/1741-2560/4/2/R01

Makeig S, Bell A, Jung T, Sejnowski T (1996) Independent component analysis of electroencephalographic data. Advances in Neural Information Processing Systems (NIPS) 8:145–151

Meier L, Van De Geer S, Bühlmann P (2008) The group lasso for logistic regression. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70(1):53–71, DOI 10.1111/j.1467-9868.2007.00627.x

Middendorf M, McMillan G, Calhoun G, Jones KS (2000) Brain-computer interfaces based on the steady-state visual-evoked response. IEEE Transactions on Rehabilitation Engineering 8(2):211–214, DOI 10.1109/86.847819

Müller K, Anderson CW, Birch GE (2003) Linear and nonlinear methods for brain-computer interfaces. IEEE Transactions on Neural Systems and Rehabilitation Engineering 11(2):165–169, DOI 10.1109/TNSRE.2003.814484

Ng AY, Jordan MI (2002) On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In: Advances in Neural Information Processing Systems, MIT Press, Vancouver, BC, pp 841–848

Nunez PL, Srinivasan R (2005) Electric Fields of the Brain: The Neurophysics of EEG, 2nd Edition, 2nd edn. Oxford University Press, USA

Nunez PL, Silberstein RB, Cadusch PJ, Wijesinghe RS, Westdorp AF, Srinivasan R (1994) A theoretical and experimental study of high resolution EEG based on surface laplacians and cortical imaging. Electroencephalography and Clinical Neurophysiology 90(1):40–57

Perrin F, Pernier J, Bertrand O, Echallier JF (1989) Spherical splines for scalp potential and current density mapping. Electroencephalography and Clinical Neurophysiology 72(2):184–187

Pfurtscheller G, Neuper C (2001) Motor imagery and direct brain-computer communication. Proceedings of the IEEE 89(7):1123–1134, DOI 10.1109/5.939829

Pfurtscheller G, Brunner C, Schlögl A, Lopes da Silva F (2006) Mu rhythm (de)synchronization and EEG single-trial classification of different motor imagery tasks.

NeuroImage 31(1):153–159, DOI 10.1016/j.neuroimage.2005.12.003
Ramoser H, Müller-Gerking J, Pfurtscheller G (2000) Optimal spatial filtering of single trial EEG during imagined hand movement. Rehabilitation Engineering, IEEE Transactions on 8(4):441446
Schölkopf B, Smola AJ (2001) Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, 1st edn. The MIT Press
Selim AE, Wahed MA, Kadah YM (2008) Machine learning methodologies in brain-computer interface systems. In: IEEE Proceedings of the Cairo International Biomedical Engineering Conference (CIBEC) 2008, IEEE, pp 1–5, DOI 10.1109/CIBEC.2008.4786106
Tomioka R, Aihara K, Müller K (2007) Logistic regression for single trial EEG classification. Advances in Neural Information Processing Systems 19:13771384
Vicente MA, Hoyer PO, Hyvarinen A (2007) Equivalence of some common linear feature extraction techniques for Appearance-Based object recognition tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(5):896–900, DOI 10.1109/TPAMI.2007.1074
Wolpaw JR, Birbaumer N, McFarland DJ, Pfurtscheller G, Vaughan TM (2002) Brain-computer interfaces for communication and control. Clinical Neurophysiology: Official Journal of the International Federation of Clinical Neurophysiology 113(6):767–791
Ye J (2006) Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. Journal of Machine Learning Research 6(1):483
Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67(2):301–320, DOI 10.1111/j.1467-9868.2005.00503.x