Online Feature Selection for Brain Computer Interfaces

Gareth Oliver
Research School of Computer Science
Australian National University
Canberra, Australia 0200
Email: Gareth.Oliver@anu.edu.au

Peter Sunehag
Research School of Computer Science
Australian National University
Canberra, Australia 0200
Email: Peter.Sunehag@anu.edu.au

Tom Gedeon
Research School of Computer Science
Australian National University
Canberra, Australia 0200
Email: Tom.Gedeon@anu.edu.au

Abstract—**Online adaptation of Brain Computer Interfaces allows for arduous training periods to be circumvented. To do this we must adapt a classifier to a new session, or better yet, a new subject. We initially outline a procedure to perform online adaptation of both the classifier's weights and the feature selection and confirm its use in session to session transfer. We found that retraining both feature selection and the classifier resulted in an average improvement of** 5% **over simply retraining the classifier, and as high as** 10%. **To avoid a retraining phase the online adaptation must be performed without labeled data. We propose and compare several methods to adapt the feature selection on unlabeled data, making use of both semi-supervised learning and interactive error potentials. From this we determined that performing a weighted feature selection performed the best, and the proposed novel approach of combining semi-supervised learning and interactive error potentials outperformed performing each individually. To improve the subject to subject adaptation when a database of previous subjects is available, we investigated using Weighted Majority Voting to weight the classifier towards subjects in that database that are useful for the new subject. We found this approach to outperform pooling all data.**

I. Introduction

The ideal Brain Computer Interface (BCI) would not require a recalibration before each session. One solution to this problem is to allow the recalibration to occur while the user is using the device. This has been done quite successfully through the use of semi-supervised learning [1] as well as simulated interactive error potentials (IErrP) [2], [3]. These methods both make use of new unlabeled data by assuming that if a condition is satisfied (classification above a certain threshold in the case of semi-supervised learning and no IErrP present in the case of IErrP), then the label given to the data is correct and it can be used for recalibration purposes.

Feature selection using multiple feature extraction methods has been met with a lot of success [4]–[6], using both wrapper and filter approaches. It has also been shown that different feature extraction methods perform better dependent on the subject. We examine the need for retraining of feature selection from session to session, and compare a variety of methods for doing so online on unlabeled data.

A recent extension of support vector machines called uncertain label support vector machines [7] allows each label to also have a certainty. In addition to exploring feature selection, we propose the use of the uncertain label SVM (uSVM) to take into account the uncertainty in the labels of data used for online retraining of a BCI. As the training data's labels are given whereas the retraining data's labels are informed guesses a strong importance should be placed on the initial data's labels. This is indeed the effect of the uSVM, and its advantage in this task.

While in many cases previous session data will be available for a subject, it would be ideal to also be able to adapt to a subject from other subjects. Weighted Majority Voting [8] allows a particular classifier to become more or less relevant to the overall classification dependent on its results so far. We propose its use when a database of subjects are available.

We will first introduce the structure of the classifier being used, as well as the feature selection method chosen. We then introduce the established methods for deciding if unlabeled data should be used for retraining, as well as proposing a combination approach. Additionally we discuss three methods for using the new data for retraining. We carried out three sets of experiments. The first is to confirm the value of session to session Feature Selection, while the second compares the different methods for online retraining and determines their viability. Finally we examine the different method's use in retraining between different subjects.

II. Classifier

In this section the components of the classifier are laid out. The components of the classifier can be broken into four segments; preprocessing, feature extraction, feature selection and classification. Each will be outlined in the subsections below.

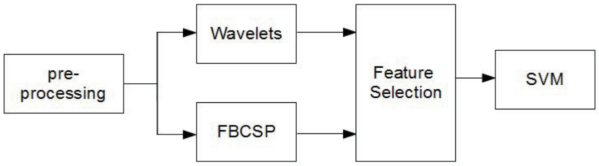

Fig. 1. Classifier Structure.

A. Preprocessing

Each data segment was preprocessed by performing channel selection to remove unnecessary channels. Additionally the data was normalised. Frequency filtering was performed dependent on the feature extraction method.

B. Feature Extraction

Two different approaches were implemented, and combined to form a single feature vector. The first, Filter Bank Common Spatial Patterns (FBCSP) [9], is an extension to Common

978-1-4673-5871-2/13/$31.00 ©2013 IEEE

Spatial Patterns [10]. The second was a pair of wavelet transforms method [11], [12]. The results of each of these are concatenated to form a single feature vector. FBCSP gives a total of 18m features, where m is the number of channels to be selected by CSP within each frequency sub-band. The wavelet methods give a total of 2*L*C features, where L is the level of decomposition and C is the number of different channels.

1) FBCSP: CSP is a spatial filtering method that reduces the dimensionality of the data so as to maximise the variance between classes. FBCSP further extends CSP to select subject specific frequency bands by using a frequency filter bank. In this experiment 9 Chebyshev Type II filters were used to decompose the signal into 9 frequency ranges. These were [4-8hz,8-12hz... 36-40hz]. CSP is then performed on each of these sub-bands. For completeness the algorithm is described below.

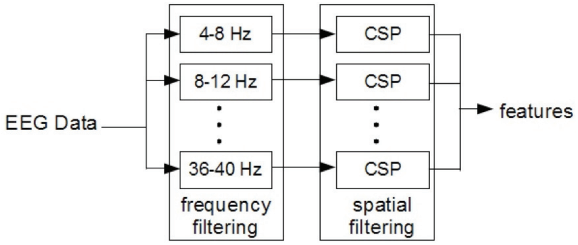

Fig. 2. FBCSP.

$$R_a = X_a . X_a^T$$
$$R_b = X_b . X_b^T$$

The co-variance matrix of each class, R_a and R_b, are estimated from the training data.

$$\begin{aligned} R &= R_a + R_b \\ &= U_0 . \Sigma . U_0^T \\ P &= \Sigma^{-1/2} . U_0^T \\ S_a &= P . R_a . P^T \\ S_b &= P . R_b . P^T \end{aligned}$$

The eigenvalues Σ and eigenvectors U_0 of R are found. This is then used to obtain the whitening matrix P. This is used to whiten the covariance matrices. This gives the whitened matrices S_a and S_b. After ranking the vectors in descending order by eigenvalue, the first m eigenvectors are selected from S_a and combined to form U_a. Similarly the first m eigenvectors of S_b are selected and combined to form U_b.

$$SF_a = U^T . P$$
$$SF_b = U^T . P$$

U_a and U_b are used to construct the corresponding spatial filter matrices SF_a and SF_b. The log of the variance of each trial is then selected as a feature. This gives a feature vector containing m values for each class, as given below for some frequency sub-band X.

$$features = (var(SF.X)) \qquad (1)$$

The features of each frequency sub-band are then concatenated together to give 18m features in total. For the purposes of these experiments m was set to 2 as in [1].

2) Wavelet Methods: Wavelet methods decompose a signal into frequency bands, from which a variety of different features can be extracted. This was introduced to BCI as Wavelet Transforms [11], and has been extended as Wavelet Packet Decomposition [12]. In this paper, Wavelet Transforms were used to decompose each signal. Wavelet transforms decompose a signal into its frequency bands by splitting a signal into its lower and higher components, then they are recursively applied to the lower frequency band. The implementation in the python library Pywavelets was used in these experiments. Two different features were extracted from the frequency sub-bands. These are the sub-band mean (MEA) and the sub-band energy (E) [12]. For these experiments the Sym17 mother wavelength was used and the data was decomposed to level 3 as they have been shown to produce good results [11].

C. Feature Selection

The goal of feature selection is, given some feature set F with k features, find the subset S $\subseteq$F with k features that minimises the classification error [13]. There are two general classes of feature selection techniques that have been used in BCI. The wrapper approach where features are selected using the classifier, and the filter approach where they are selected independent of the classifier. The wrapper approach is often more effective than filter approaches as it takes into account the classifier's ability to separate the features. However filter approaches are generally significantly faster to perform [13]. In this paper Mutual Information (MI) was used to perform feature selection. MI algorithms restate the feature selection goal as given some feature set F with k features, find the subset [6] S $\subseteq$ F with k features that maximises the MI. MI between two random variables can be calculated as

$$I(X; Y) = H(Y) - H(Y|X) \qquad (2)$$

where the entropy H(X) of a d-dimensional random variable X is

$$H(X) = -\sum_{x} p(x) \log_2 p(x) \qquad (3)$$

and the conditional entropy of the random variables X and Y is

$$H(Y \mid X) = -\sum_{x,y} p(x, y) \log_2 p(y \mid x) \qquad (4)$$

In the above p is the probability function. For the classification problem the features are continuous while the classes are discrete. This leads to the MI being calculated between some input features X and class Ω as

$$I(X; \Omega) = H(\Omega) - H(\Omega|X) \qquad (5)$$

where

$$H(\Omega|X) = -\int_{X} \sum_{\omega} p(\omega|x) \log_2 p(\omega|x) dx \qquad (6)$$

In this paper a Filter Approach was taken, as the goal of online feature adaptation requires the feature selection to be

```
Inputs

F ←— $f_1$,$f_2$..$f_d$
S ←— ∅
MI[D] −empty array for MI
k −desired features

Algorithm
for (i = 0,$i < d$, i++) do
  MI[i] = I(F$_d$)
end for
for (j=0,$j < k$ , j++) do
  $i$ ←— argmax(MI)
  $S$ ←—S ∪$f_i$
  $F$ ←—F\$f_i$
  MI[i] = −∞
end for
return  S
```

Fig. 3. MIBIF Algorithm.

faster. A Parzen Window was used to estimate $p(\omega|X)$. As examining all possible feature subgroups is too computationally expensive some form of greedy algorithm are usually used. The commonly used Mutual Information Based Individual Feature (MIBIF) was used in this paper. The algorithm is described in Alg 3 [9]. In it, the MI between each feature and the classes is calculated and then the k features with the highest MI are selected. When retraining, the new data was weighted so as to have higher weight than the initial training data. This redefines the MI as

$$\mathrm{I}(f_j;\omega) = w_1 \mathrm{I}_1(f_j;\omega) + w_2 \mathrm{I}_2(f_j;\omega) \qquad (7)$$

where $w_1, w_2 \in R$ with $w_1 < w_2$ and I_1 calculates the MI based on the previous session data while I_2 calculates MI based on the data obtained so far.

D. Weighted Majority Voting

We examined three different methods for retraining when a classifier was initially trained on another subject's data. The first approach was to train a single classifier with the combined data from all subjects. The remaining approaches used weighted majority voting [8], [14]. The weighted majority voting (WMV) algorithm trains n different classifiers, each on a different subject's data. The weighted combination of the output of each of these classifiers gives the decision of the master classifier. These methods were only used when IErrPs were available. In the conservative WMV method, when an IErrP was detected, the different classifiers which agreed with the overall classification would have their weights updated by

$$w^{t+1} = \beta w^t \qquad (8)$$

where $0 < \beta < 1$.

For the aggressive WMV method, when an IErrP is not detected the classifiers that did not agree with the overall classification are still having their weights updated by Eq. 8. This makes it closer to a Bayesian predictor [14]. The algorithm is given in Alg. 4.

E. Classification

Support Vector Machines (SVMs) are among the most popular and successful classifiers. They have been used to great success for classification of motor tasks. SVMs seek to maximise a decision boundary between the classes being classified. For a traditional SVM we define the training dataset $(x_i, l_i)_{i=1..n}$ where $x_i \in X$ and $l_i \in \{-1, 1\}$. Here X is the feature space while l_i are the labels. The objective function for the soft margin SVM can be written as [15]

$$\min_{w,\xi,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \qquad (9)$$

subject to $l_i(w.x_i + b) \geq 1 - \xi_i$ and $\xi_i \leq 0$, $i = 1..n$. One method to assign a probability p_i for an SVM classification is by using the formula

$$p_i = \frac{1}{1+e^{-a(w^T x_i + b)}} \qquad (10)$$

[7] propose a method to add additional training data where there is uncertainty in the labels. The training data set is extended to include $(x_i, l_i)_{i=n+1..m}$ where $x_i \in X$ and $l_i \in [0, 1]$ is the probability x_i belongs to class 1. The objective function is changed to

$$\min_{w,\xi,\xi^+,\xi^-,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i + \tilde{C}\sum_{i=n+1}^{m}\xi_i^- + \xi_i^+ \qquad (11)$$

In addition to the previous constraints ($l_i(w.x_i + b) \geq 1 - \xi_i$ and $\xi_i \leq 0$, $i = 1..n$), it is also subject to $z_i^- - \xi_i^- \leq w^T x_i + b \leq z_i^+ \xi_i^+$, $0 \leq \xi_i^-$ and $0 \leq \xi_i^+$ for $i = n + 1..m$. Here z_i^- and z_i^+ are defined as

$$z_i^- = -\frac{1}{a}\ln\frac{1}{p_i - \eta} - 1 \qquad (12)$$

and

$$z_i^+ = -\frac{1}{a}\ln\frac{1}{p_i - \eta} + 1 \qquad (13)$$

where

$a = \ln(1/\eta - 1)$

and η is the sum of the confidence in the labeling and the precision of the labeling. This forces the assigned probability of a data point to remain within a distance η from the given certainy for the uncertain points.

In Eq. 11, C and $\tilde{C}$ are positive real numbers which weight the importance of the uncertain labels portion with that of the true labels. In this paper we weighted them equally.

III. Unlabeled Data

An ideal BCI would not require a retraining period before

```
Inputs

β ∈ [0..1]
x − current datasegment
C − n subclassifiers, returning class label y ∈ {-1, 1}
W − n weightings of the subclassifiers
A − true if aggressive, false if conservative
IErrP − true if an IErrP is detected

Algorithm
y ← sign   nWi Ci (x)
if (I ErrP (x, y))  then
    for (i = 0, i < n, i + +) do
        if (Ci (x) == y) then
            Wi = βWi
        end if
    end for
else
    if (A) then
        for (i = 0, i < n, i + +) do
            if (Ci (x) j= y) then
                Wi = βWi
            end if
        end for
    end if
end if
```

Fig. 4. WMV Algorithm.

each session. One approach to alleviate this is to make use of the unlabeled data while the BCI device is being used. There are two approaches that have been explored in literature for this, Semi-Supervised Learning and Interactive Error Potentials. In addition we propose a new method that combines the two. Three different methods for retraining the features selected will follow that. Furthermore we extend each for use by the uSVM.

A. Semi-Supervised Learning

Semi-Supervised Learning makes use of the already trained classifier to determine if a data segment X which has been classified with label Y should be used to adapt the classifier [1]. If the label Y for the data segment X satisfies the equation

$$p(\mathrm{Y}|\mathrm{X}) > \text{threshold} \tag{14}$$

then it is used. For our classifier we define $p(\mathrm{Y}|\mathrm{X})$ as

$$p(\mathrm{Y}|\mathrm{X}) = (\varphi(\mathrm{X}, \mathrm{Y}), w) - (\varphi(\mathrm{X}, \mathrm{Y'}), w) \tag{15}$$

Where Y' is the other class to Y. By only adding data points that satisfy a certain threshold, it reduces the chance of adding a misclassified segment.

B. Interactive Error Potentials

Error Potentials present a unique way to verify whether a decision by the BCI is correct or not without increasing the user's cognitive load. Responses in the fronto-central electrodes have been known to occur when a subject makes an error, or observes an error being made. Interactive Error Potentials (IErrPs) define the class of Error Potentials where the user observes the result of an error made by the BCI [16]. These have been used to automatically error correct [2], and more recently their use for gaining additional data has been explored [3]. In [3] the probability of detecting an IErrP given correct and incorrect classes was estimated and used to simulate IErrP. In this paper we make use of their results saying $p(\text{correct}|\text{correct}) = 88\%$ and $p(\text{incorrect}|\text{incorrect}) = 56\%$ to simulate the effect of IErrP. Labels where no IErrP are detected are assumed to have the correct label and that segment X and label Y are used for retraining. Due to the low probability of p(incorrect|incorrect), an IErrP being detected does not imply that another class label is the correct label with any degree of certainty. The algorithm is given in Fig.5.

Given real IErrP it would be possible to use the confidence of the classifier detecting the IErrP to define the certainty of the label. In this simulated case we set the confidence to 88% if no error is detected and 44% if it is. These correspond to $p(\text{correct}|\text{correct})$ and $p(\text{correct}|\text{incorrect})$ respectively.

C. Hybrid Approach

The above two methods can be combined to both increase the amount of data segments that can be used for training and the certainty of the labels. The acceptance of a data segment X with label Y can be defined as

$$p(\mathrm{Y}|\mathrm{X},\mathrm{IErrP}) > \begin{cases} (\mathrm{IErrP} = \text{correct}) : \text{threshold}_1 \\ (\mathrm{IErrP} = \text{incorrect}) : \text{threshold}_1 \end{cases} \tag{16}$$

where $\text{theshold}_1 >> \text{threshold}_2$. When $\text{threshold}_1 = 0$ and $\text{theshold}_2 = \infty$ this is equivalent to the IErrP case. Similarly when $\text{threshold}_1 = \text{threshold}_2$ this becomes the Semi-Supervised Learning case. The algorithm is in Fig. 6.

The certainty of in the IErrP and the semi-supervised learning can be combined so as to result in an overall

$$\text{certainty}(\text{hybrid}) = \frac{\text{certainty}(\text{semi-supervised}) + \text{certainty}(\text{IErrP})}{2} \tag{17}$$

D. Adaptation of Feature Selection

We investigate several approaches to making use of new data to adapt the feature selection. The simplest approach is to add the new data to the training data. This can be defined as

$$\text{data}_{t+1} = \text{data}_t \cup (\mathrm{X}, \mathrm{Y}) \tag{18}$$

where (X,Y) is the new data segment and label being added and data is the total data used for calculating the features selected and training the SVM. In this setting, data_0 is the previous session's data.

```
Inputs

Y − Label
X − Current Data Segment
Y* − Correct Label
Data − Data used for Training

Algorithm
R ← Random [0.0..1.0]
if (Y == Y*) then
    if (R < 0.88) then
        Data ← Data $\cup (X, Y)$
    end if
else
    if (R > 0.88) then
        Data ← Data $\cup (X, Y)$
    end if
end if
return Data
```

Fig. 5. IErrP Algorithm for obtaining new data.

The second approach taken was to build a new feature subset from the accumulated new data. The advantage of this method is that it would not be biased by the previous session's data, with the significant disadvantage of having less total data to work with. The update rule follows Eq.18, however $\text{data}_0 = \emptyset$. The final approach taken was to apply a weight that decides the importance of individual data segments when calculating the mutual information, thereby increasing the importance of the newly acquired data for calculating the feature subset. The MI is defined as

$$\mathrm{I}\left(f_j;\omega\right)=w_1\mathrm{I}_1\left(f_j;\omega\right)+w_2\mathrm{I}_2\left(f_j;\omega\right) \quad (19)$$

where $w_1, w_2 \in R$ with $w_1 < w_2$ and I_1 calculates the MI based on the previous session data while I_2 calculates MI based on the data obtained so far.

IV. Experiments

Several experiments were designed to evaluate the proposed methods. The first set was conducted to determine whether improvement could be achieved when performing feature selection using the current session's data when compared with using an older sessions data. The second set examines the comparative success of each of the proposed methods for retraining the feature selection from session to session. Finally the third set examines whether different subjects can be used as to create the initial classifier. All of the experiments where performed using BCI competition IV dataset 2a [17]. For simplicity only two of the available four classes where used (the left and right hands). Dataset 2a consists of 9 subjects, each with two sessions of data. Each session consists of 72 synchronous trials of each task, each trial 4 seconds in duration. The data is made up of 22 electrode channels down sampled to 250hz.

```
Inputs

Y −Label
X −Current Data Segment
Y* −Correct Label
Threshold$_1$ −Rejection threshold Threshold$_2$ −
Acceptance threshold Data −Data used for Training
SVM −Trained SVM, returns class score
IErrP −Function to determine if an IErrP is detected

Algorithm
Score ←—SVM(X, Y) −SVM(X, Y')
if (IErrP(Y, Y*) then
  if (Score > threshold$_1$) then
     Data ←—Data $\cup(X, Y)$
  end if
else
  if (Score > threshold$_2$) then
     Data ←—Data $\cup(X, Y)$
  end if
end if
return Data
```

Fig. 6. The Hybrid Algorithm for obtaining new data.

A. Session to Session Feature Selection

We take the data from each subject and label it S_1 and S_2 for session one and two respectively. 10% of the trials from S_2 are randomly selected and placed in S_{2test}, while the remainder are placed in S_{2train}. S_1 is similarly partitioned with the 90% portion labeled S_{1train} while the remainder are not used in this iteration. S_{1train} and S_{2train} are then used to find two feature subsets, f_1 and f_2. Two classifiers, C_1 and C_2, are then trained using S_{2train} with their respective feature subsets. The features selected are then compared and the performance of C_1 and C_2 are evaluated on S_{2test}. This was then repeated for the 10 other 10% partitions to obtain an average accuracy and determine the difference in features selected. The entire experiment was repeated 10 times and the results averaged.

	C_1 (%)	C_2 (%)	Diff. Features
Subject 1	65	70	4
Subject 2	56	60	3
Subject 3	69	75	5
Subject 4	63	69	4
Subject 5	57	63	2
Subject 6	54	64	6
Subject 7	70	76	3
Subject 8	68	74	4
Subject 9	62	68	3
Average	63	69	4

TABLE I

Session to session feature selection comparison.

The results in Tbl. I show that an average of 4 features were selected differently between the two features, resulting in an average accuracy increase of 6% when S_{2train} was used to select the features. To confirm that these results are significant the same experiment was carried out, with S_1 and S_2 being two halves of a single session.

	Diff. Accuracy (%)	Diff. Features
Subject 1	2	1
Subject 2	1	2
Subject 3	3	2
Subject 4	1	1
Subject 5	0	1
Subject 6	2	2
Subject 7	3	3
Subject 8	1	2
Subject 9	2	2
Average	2	1

TABLE II
WITHIN SESSION COMPARISON.

As it can be seen in Tbl. II, the difference between the features selected and the classification accuracy of C_1 and C_2 are significantly lower than when two different sessions were used, so we conclude that performing feature selection on the current session can improve the results.

B. Session to Session Online Feature Selection

The second experiment once again split the data into S_1 and S_2 and the order of the trials in S_2 were randomised. S_1 was used for initial feature selection and training to create an initial classifier C_1. C_1 was used to classify the first 10 data segments of S_2, which were then used to retrain C_1 to C_2. This was repeated for the next 10 data segments until all were used. Four different methods were used for determining if a classified data segment should be used for retraining. These were Supervised, where the correct label was always known, Semi-Supervised Learning, IErrP and the Hybrid Semi-Supervised Learning and IErrP. Additionally the three different methods for retraining the features selected: adding, new feature sub-set and weighted adding were used. Finally the uSVM was performed using the weighted adding method for semi-supervised, IErrP and Hybrid methods. This gave a total of 15 different methods. The experiment was run 10 times for each subject to get a more consistent result, and the average is given in Figure 7 and Tbl III .

	Ave. Accuracy (%)
Supervised: adding	63.2
Supervised: new subset Supervised:	62.7
weighted adding	64.7
Semi-Supervised: adding Semi-	61.0
Supervised: new subset	51.9
Semi-Supervised: weighted adding	62.3
Semi-Supervised: Session: uSVM	63.5
IErrP: adding	61.9
IErrP: new subset	57.8
IErrP: weighted adding	62.5
IErrP: Session:	63.4
uSVM Hybrid: adding	62.0
Hybrid: new subset	55.0
Hybrid: weighted adding	62.7
Hybrid: Session: uSVM	64.5

TABLE III
SESSION TO SESSION AVERAGE ACCURACIES.

The results show that, in the supervised case the new feature subset method reaches the highest accuracy, however it starts with the lowest accuracy and has a lower overall average. Additionally, the weighted approach outperformed the adding method. Using IErrP to determine whether a data segment should be used outperformed the Semi-Supervised learning approach and it allowed the new feature sub-set method to improve despite its low initially accuracy. Finally the combination of Semi- Supervised Learning and IErrP method performed better than IErrP or Semi-Supervised Learning by themselves for the adding or weighted methods, although the IErrP method was better for the new feature sub-set method. Over each of the methods the weighted method continued to outperform the adding method. In each case, the uSVM outperformed the comparative traditional SVM method.

From these results we concluded that, by using the combination of IErrP and Semi-Supervised Learning methods, the data can be used to retrain feature selection on unlabeled data. Additionally the weighted method is superior in most cases for unlabeled data. We also concluded that the uSVM was an improvement over the traditional SVM. Finally the success of the IErrP method implies that it could be used to improve a classifier with little initial data.

C. Subject to Subject Online Feature Selection

We conducted the final experiment to consider cases where no previous session data is available for the subject while other subject's data is. We compared both the effectiveness of the uSVM with the traditional SVM as well as three different approaches to performing retraining using previous subject's data when IErrP were available. S_2 is once again defined as the current subject's session 2 data. S_1 is set as the other 8 subject's session 2 data and used to train an initial C_1 as before. In the IErrP and hybrid methods both WMV methods were used in addition to the basic combined method. The results can be seen in the graphs in Figure 8 and the average accuracies can be seen in IV.

	Ave. Accuracy (%)
Semi-Supervised: combined Subject: SVM	56.7
Semi-Supervised: combined Subject: uSVM	56.5
IErrP: combined Subject: SVM	59.5
IErrP: combined Subject: uSVM	58.5
IErrP: cWMV Subject: SVM	60.0
IErrP: cWMV Subject: uSVM	59.1
IErrP: aWMV Subject: SVM	61.3
IErrP: aWMV Subject: uSVM	60.2
Hybrid: combined Subject: SVM	60.1
Hybrid: combined Subject: uSVM	59.2
Hybrid: cWMV Subject: SVM	60.2
Hybrid: cWMV Subject: uSVM	59.5
Hybrid: aWMV Subject: SVM	61.8
Hybrid: aWMV Subject: uSVM	60.5

TABLE IV
SUBJECT TO SUBJECT AVERAGE ACCURACIES.

Unsurprisingly the subject to subject transfer performed worse than the session to session. Unlike the previous experiments however, uSVM was worse than traditional SVM. We believe that this is due to the increased uncertainty

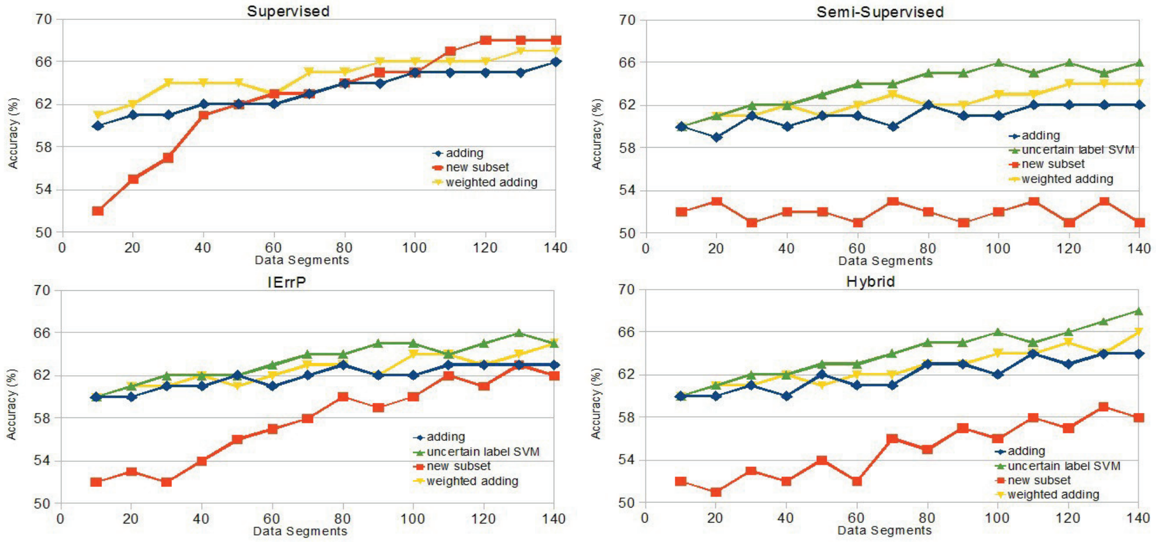

Fig. 7. Online adaptation from session to session.

in the S_1 data as training data for S_2. While the true labels are known, due to the difference in different subjects, the original data is likely to be less reliable than the new data. To some extent a similar problem may be occurring in the session to session experiments. One potential solution to this would be to assign a certainty < 1 to the labels for the data of S_1 to account for this, however our attempts were unable to produce a result better than that of the traditional SVM for the subject to subject case.

We additionally found that, in cases where WMV methods could be used rather than simply combining the data it proved more effective for both the traditional SVM and the uSVM. Additionally we found that the aggressive WMV was consistently more effective than the conservative WMV in both the hybrid and the IErrP methods.

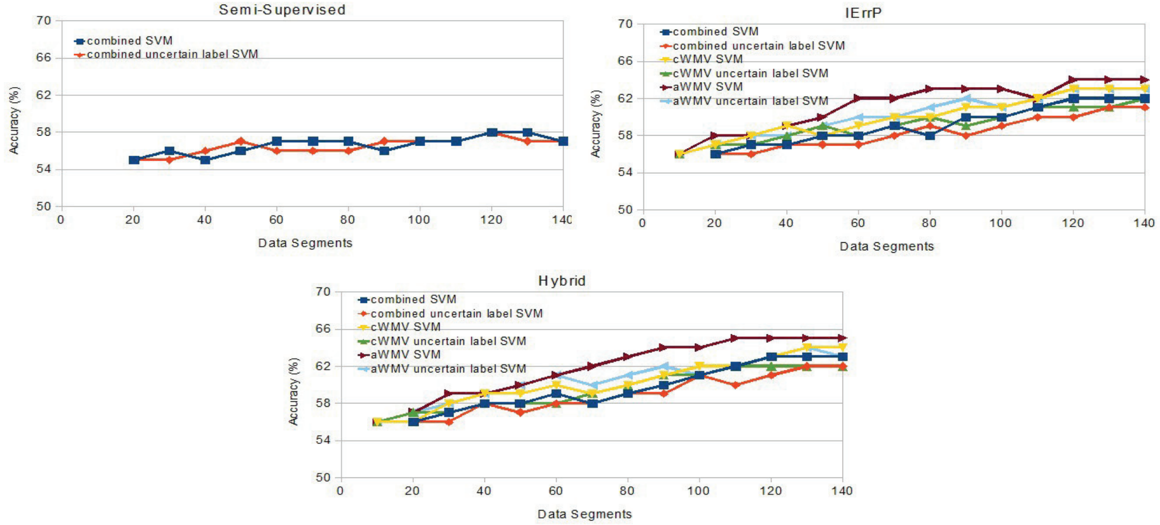

Fig. 8. Online adaptation from subject to subject.

V. Conclusion

Our initial set of experiments demonstrated the potential value of adapting feature selection in addition to retraining the weights of the classifier. We were also able to successfully perform session to session adaptation without the use of new labeled data, with our novel hybrid method outperforming the others. Additionally, we found the uncertain label SVM outperformed the traditional SVM in adapting from session to session, but it failed to do so when adapting between subjects. Finally, we found that the Weighted Majority Voting approach to adapting between multiple subjects was highly successful, with the aggressive approach performing best.

References

[1] K. K. Ang, Z. Y. Chin, H. Zhang, and C. Guan, “Filter bank common spatial pattern (fbcsp) algorithm using online adaptive and semi-supervised learning,” in *Neural Networks (IJCNN), The 2011 International Joint Conference on*, 31 2011-aug. 5 2011, pp. 392 –396.

[2] P. Ferrez and J. del R. Millan, “Error-related eeg potentials generated during simulated brain 2013;computer interaction,” *Biomedical Engineering, IEEE Transactions on*, vol. 55, no. 3, pp. 923 –929, march

2008.

[3] X. Artusi, I. K. Niazi, M.-F. Lucas, and D. Farina, “Performance of a simulated adaptive bci based on experimental classification of movement-related and error potentials,” *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 1, no. 4, pp. 480 –488, dec. 2011.

[4] G.-z. Yan, T. Wu, and B.-h. Yang, “Automated feature selection based on an adaptive genetic algorithm for brain-computer interfaces,” in *Simulated Evolution and Learning*, ser. Lecture Notes in Computer Science, T.-D. Wang, X. Li, S.-H. Chen, X. Wang, H. Abbass, H. Iba, G.-L. Chen, and X. Yao, Eds. Springer Berlin / Heidelberg, 2006, vol. 4247, pp. 575–582, 10.1007/1190369773. [Online]. Available: http://dx.doi.org/10.1007/1190369773

[5] N. Dias, L. Jacinto, P. Mendes, and J. Correia, “Feature down-selection in brain-computer interfaces,” in *Neural Engineering, 2009. NER ’09. 4th International IEEE/EMBS Conference on*, 29 2009-may 2 2009, pp. 323–326.

[6] M. Deriche and A. Al-Ani, “A new algorithm for eeg feature selection using mutual information,” in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP ’01). 2001 IEEE International Conference on*, vol. 2, 2001, pp. 1057 –1060 vol.2.

[7] E. Niaf, R. Flamary, C. Lartizien, and S. Canu, “Handling uncertainties in svm classification,” in *Statistical Signal Processing Workshop (SSP), 2011 IEEE*, june 2011, pp. 757 –760.

[8] N. Littlestone and M. Warmuth, “The weighted majority algorithm,” in *Foundations of Computer Science, 1989., 30th Annual Symposium on*, oct-1 nov 1989, pp. 256 –261.

[9] K. K. Ang, Z. Y. Chin, H. Zhang, and C. Guan, “Filter bank common spatial pattern (fbcsp) in brain-computer interface,” in *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, june 2008, pp. 2390–2397.

[10] S. Lemm, B. Blankertz, G. Curio, and K.-R. Muller, “Spatio-spectral filters for improving the classification of single trial eeg,” *Biomedical Engineering, IEEE Transactions on*, vol. 52, no. 9, pp. 1541–1548, sept. 2005.

[11] J. Sherwood and R. Derakhshani, “On classifiability of wavelet features for eeg-based brain-computer interfaces,” in *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, june 2009, pp. 2895–2902.

[12] W. Ting, Y. Guo-zheng, Y. Bang-hua, and S. Hong, “Eeg feature extraction based on wavelet packet decomposition for brain computer interface,” *Measurement*, vol. 41, no. 6, pp. 618 – 625, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0263224107000711

[13] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944968

[14] N. Cesa-Bianchi, D. P. Helmbold, and S. Panizza, “On bayes methods for on-line boolean prediction,” *Algorithmica*, vol. 22, pp. 112–137, 1998, 10.1007/PL00013825. [Online]. Available: http://dx.doi.org/10.1007/PL00013825

[15] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[16] P. Ferrez and J. del R. Millan, “Error-related eeg potentials generated during simulated brain computer interaction,” *Biomedical Engineering, IEEE Transactions on*, vol. 55, no. 3, pp. 923 –929, march 2008.

[17] B. Blankertz, G. Dornhege, M. Krauledat, K.-R. Muller, and G. Curio, “The non-invasive berlin braincomputer interface: Fast acquisition of effective performance in untrained subjects,” *NeuroImage*, vol. 37, no. 2, pp. 539 – 550, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811907000535