TRANSFER LEARNING FOR EEG BASED BCI USING LEARN++.NSE AND MUTUAL INFORMATION

Matthew Sybeldon, Lukas Schmit, Ervin Sejdic, Murat Akcakaya

University of Pittsburgh, Department of Electrical and Computer Engineering

ABSTRACT

In this paper, the use of mutual information and the Learn++.NSE algorithm is proposed to create an EEG SSVEP BCI system that can select and utilize data sets originating from a group of users. In typical BCI systems, the nonstationarity in the EEG prevents the system from blindly applying training data from other users to the incoming data. Mutual information is introduced to select previous data sets that provide the most information about current random variables. A signed rank test was employed to show that this configuration outperformed both normal Learn++.NSE ensembles and LDA classifiers. This indicates that mutual information and ensemble learning techniques may prove useful in improving user transferability in SSVEP systems with low computational requirements.

Index Terms— EEG, BCI, SSVEP, Transfer Learning, Mutual Information

1. INTRODUCTION

Brain-Computer Interfaces (BCI) are a promising input modality for users who are unable to interact with a computer by traditional means due to injury. Electroencephalography (EEG) signals are well suited toward BCI applications due to their high temporal resolution and lower cost relative to other commonly used signals extracted from the brain [1]. In practice, these systems are hampered by nonstationarities in the statistical distributions calculated from EEG features [1]. This requires each user to undergo a lengthy and tiresome system calibration for every usage session. Transfer learning has been proposed as a method to reduce the calibration requirements [2] [3] [4].

One early effort formulated an ensemble of classifiers and calculated the accuracy of that classifier on an incoming data set with binary training labels. Decisions were based on a linear combination of voting weights and classifier decisions [2] [3]. This option is attractive for simplicity, but it has issues in large ensembles if a large amount of classifiers perform at chance levels. A more robust weighting scheme is needed for transfer learning with an ensemble.

Another ERP study used a combination of unsupervised Bayesian learning, Expectation Maximization (EM), and transfer learning [4]. One drawback to this system is the high computational requirements due to the use of Bayesian inference in conjunction with an EM-like algorithm. A more computationally feasible solution is desired, particularly for signals such as SSVEP, which is generally chosen for relative simplicity [5].

One might observe from currently published research that there is a gap in the literature with regards to inter-user transferability of SSVEP BCI systems due to perceived simplicity. Currently, canonical correlation analysis (CCA) is considered state of the art for SSVEP systems for its high information transfer rates (ITR) and low/zero calibration requirements [6]. These systems generate correlation scores between a group of reference signals and choose from the maximum score. There are counterexamples in our own collected data and in the general literature illustrating that this method faces difficulties due to its assumption that each frequency's correlation score is impacted uniformly by the stimuli [7]. A possible solution is a normalized score by dividing by neighboring frequencies' correlation score [8]. If the SNR is not constant across the spectrum, classification using these scores is required, which does not hold across users.

We now present an example of an inter-user nonstationarity that was encountered in our collected data. Figure 1 shows the average power measurement in a system where two flickering stimuli at 6 Hz and 20 Hz were shown. The first two harmonics of each stimulus frequency were measured. Despite attention being focused on the 20 Hz stimulus, participant 1 had a higher power in the 6 Hz measurement whereas participant 2 responded to the 20 Hz as predicted. This prevents a classifier trained on one participant from being applied to the other. As such a transfer learning mechanism with low computational complexity must be developed for SSVEP systems.

In this paper, an SSVEP system is proposed that utilizes an existing ensemble learning algorithm designed to handle nonstationarities among data sets. Our contribution is twofold: (1) The use of a nonstationary ensemble learning algorithm toward an SSVEP BCI system; and (2) introduction

Funding and support provided by the University Of Pittsburgh Swanson School Of Engineering, the Office of the Provost, and the University of Pittsburgh Central Research Development Fund.

Lukas Schmit is a senior in The Gregory School, and he contributed this work while enrolled in the University of Pittsburgh Cancer Institute Summer Internship Program.

978-1-5090-4117-6/17/$31.00 ©2017 IEEE

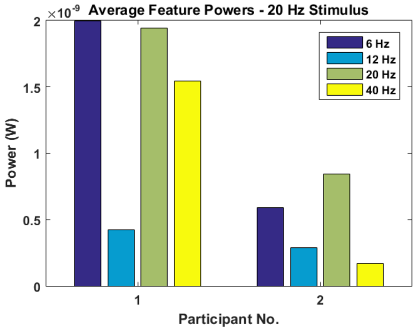

Fig. 1: An example of nonstationarity in the average power measurements between two users in an SSVEP study.

of a transfer learning approach for data selection that utilizes mutual information to populate the ensemble with data sets recorded from multiple participants. More specifically, thirty data sets generated by ten users were used to classify incoming data. The data sets are used to form an ensemble of classifiers as dictated by Learn++.NSE where the oldest ensemble member is designated as the data set with the least mutual information. Test data was then used to evaluate the ensembles' performance. The ensembles formed through the proposed transfer learning approach were found to have better accuracies than a traditional Learn++.NSE ensemble utilizing all data sets belonging to a specific participant and than an LDA classifier trained on the most recent data set.

2. METHODS

2.1. Learn++.NSE

Learn++.NSE was chosen as the ensemble learning algorithm due to its ability to assign useful weights for ensembles of any size while keeping computational complexity low. The details of this algorithm are summarized in Algorithm 1 [9].

First, define an ensemble hypothesis for a given data point at a discrete time *t* as $H^t(x)$. Voting weights V^t for each of the k^t ensemble members must be found.

$$V^t = [V_1^t, V_2^t, \ldots, V_{k^t}^t] \tag{10}$$

Each of the k^t individual member hypotheses $h_{k^t}^t(x)$ will generate up to *c* candidate decisions for the entire ensemble. The final ensemble hypothesis $H^t(x)$ is chosen such that:

$$H^t(x) = \arg\max_c\left(\sum_{i=1}^{k^t}(h_i^t(x) == c) * V_i^t\right) \tag{11}$$

Next, a data weight distribution w^t for the incoming training data set is defined. The distribution is initialized uniformly, so $w^t(i) = \frac{1}{m^t}$, where m^t is the amount of training data points in newly available data set D^t.

First, the ensemble error rate, E^t, is assessed on the data set D^t. This is done using the previous ensemble hypothesis $H^{t-1}(x)$ from k^{t-1} member classifiers on each data point

```
Data: Data set $D^t$ of length $m^t$
A designated base classifier algorithm
Real valued (a,b) sigmoid parameters
Ensemble hypotheses $H^t(x_i^t) = \hat{y_i^t}$ with size $k^t$
Result: Trained ensemble $H^t$
for t = 1, 2, 3, ... do
    if t = 1 then
        Initialize weight vector $w^t(i) = \frac{1}{m^t}$ and go to
         step 4
    end
    1. Determine ensemble error for current data set $D^t$
        $E^t = \frac{1}{m^t}\sum_{i=1}^{m^t} \hat{y}_i^t \neq y_i^t$    (1)
     2. Perform boosting step if correctly classified
        $w^t(i) = \frac{1}{m^t} * E^t$    (2)
     Otherwise
        $w^t(i) = \frac{1}{m^t}$    (3)
     3. Normalize $w^t$ so that $w^t$ is a distribution
    4. Train new base classifier on $D^t$
    5. Compute individual classifier errors on $D^t$ for
     k=1:$k^t$
        $\epsilon_k^t = \sum_{i=1}^{m^t} w^t(i) * (h(x_i^t) \neq y_i^t)$    (4)
    If $\epsilon_k^t > \frac{1}{2}$ for $k < k^t$, set $\epsilon_k^t = \frac{1}{2}$
    If $\epsilon_k^t$ = $\frac{1}{2}$ for $k = k^t$, retrain latest classifier
    6. Normalize individual classifier errors
        $\beta_k^t = \frac{\epsilon_k^t}{1 - \epsilon_k^t}$    (5)
     7. Compute weighted average of all classifier
     errors using sigmoidal curve
        $\omega_k^t = \frac{1}{1 + e^{-a(t-k-b)}}$    (6)
        $\omega_k^t = \frac{\omega_k^t}{\sum_{j=t-k}^{t} \omega_k^t}$    (7)
        $\bar{\beta}_k^t = \sum_{j=0}^{t-k} \omega_k^{t-j} * \beta_k^{t-j}$    (8)
    8. Calculate voting weights
        $V_k^t = \log(\frac{1}{\bar{\beta}_k^t})$    (9)
end
```

Algorithm 1: Outline of the Learn++.NSE algorithm

x in D^t. Each of the i data points in D^t is assigned a new weight $w^t(i)$ by multiplying its current weight by the ensemble error rate E^t if the data point was classified correctly by the ensemble. Since $E^t \leq 1$, correctly classified points will always have a lower weight in the distribution. These steps are represented by steps 2 and 3 in Algorithm 1.

Learn++.NSE handles nonstationarities by adding a new hypothesis $h^t_{k^t}(x)$ on the most recent training data set D^t and by calculating voting weights V^t of the resulting ensemble. The voting weights are found by evaluating the individual classifier error rate ϵ^t_k for each of the k^t classifiers as shown by step 5. These error rates are also affected by the data weight distribution D^t. Note that the age of the ensemble members is not directly taken into account.

A sigmoidal error weighting scheme is included in step 7 to prevent overfitting to the data [9]. The sigmoid curve weight before normalization, $\omega(t)$, is defined by:

$$\omega^t_k = \frac{1}{1 + e^{-a(t-k-b)}} \quad (12)$$

In this formula, k is the classifier position within the ensemble. The quantity $t - k$ is the time difference between the current time and the classifier creation time. Two parameters a and b are also introduced. These control the slope and horizontal offset respectively. These hyperparameters need to be tuned according to the data [10]. This was accomplished using a grid search in the hyperparameter space using ten fold cross validation for testing, with nine fold internal cross validation for every point in the hyperparameter space.

The final classifier voting weight of classifier k, V^t_k, is based on a combination of the errors from the current and past data sets. The weighted error rate $\bar{\beta}^t_k$ is calculated based on the procedures shown in step 7.

The final voting weights for the k-th classifier, are obtained by taking the log reciprocal of the $\bar{\beta}^t_k$ [2].

2.2. Incorporating Mutual Information

Mutual information is a measure of how much information one random variable provides about another. In this experiment, mutual information was used as a method for finding which previously collected data sets best represented the incoming data set. In general, the mutual information between vector random variables X and Y is defined as [11]:

$$\int_X \int_Y p(X,Y) \log(\frac{p(X,Y)}{p(X)p(Y)}) dY dX \quad (13)$$

Applying a Gaussian distribution assumption, the mutual information between X and Y of equal dimension with covariance matrices C_X and C_Y respectively can be calculated as:

$$I(X:Y) = \frac{1}{2} \log(\frac{det(C_X)det(C_Y)}{det(C)}) \quad (14)$$

The covariance matrix C is the full covariance matrix obtained by concatenating X and Y. If a data set contains n vectors for each variable, then there are n^2 combinations that will yield their own unique estimates of C. Averaging these estimates will reduce the overall estimate variance of C.

The mutual information can be incorporated into Learn++.NSE by receiving a new training data set. From that data set, the true class labels can be used to calculate the posterior probability distributions for each class. The total mutual information between every pre-existing data set and the incoming data set is found. From there, the m highest ranking data sets are chosen for training in the Learn++.NSE framework where the lowest ranking data set is introduced first, thereby making it the oldest classifier in the ensemble.

3. EXPERIMENTS

System Description: We developed an SSVEP-based BCI for binary selection that employed two flickering checkerboards at 6 and 20 Hz. The system was realized on a Lenovo ThinkPad laptop running 64-bit Windows 7. MATLAB 2015a was used for data acquisition, signal processing, feature extraction and classification; and Psychtoolbox (a freely available toolbox for creating time-accurate stimuli for experiments) was used for presentation.

The system was connected to a g.Tec g.USBamp via USB for data acquisition. The amplifier was connected to a g.gammaBox which was directly connected to the electrodes. Single channel EEG was used over the visual cortex (OZ on the 10-20 system) with a butterfly electrode. A ground electrode was placed over the forehead (FPZ on the 10-20 system). A reference electrode was clipped to the earlobe. A parallel port cable was also used to output digital values to the amplifier depending on the system's current state. This digital value was sampled alongside the EEG data to easily separate the EEG data of interest.

Participant Description and Experimental Procedures: Ten healthy participants (8 males and 2 females) were enrolled in this study according to the University of Pittsburgh IRB No. PRO15060140. All participants were required to be at least 18 years of age and have no history of epilepsy.

All participants were asked to direct their covert attention randomly at one of the two checkerboards at the start of every trial. Each trial consisted of flickering of the checkerboards for five seconds. In one usage session one hundred trials were presented. There were three usage sessions. In all sessions, a calibration phase was taken to collect training data. On the final session, a test phase of equal length was used to collect testing data where the ensemble would be evaluated.

Signal Processing and Feature Extraction: The EEG segments collected during each trial were sampled at 256 Hz and filtered using a 150th order constrained least squares FIR filter from 2-45 Hz [12]. A power spectral density estimate was obtained using Welch's method [13]. Features were made using the first two harmonics of the stimuli frequencies to obtain a four dimensional feature vector.

Classification: A linear discriminant classifier was used in the Learn++.NSE ensemble to reduce computational complexity. For each participant, ensembles were formed using groups of three individual classifiers. Two groups of ensembles were generated. The first group consisted of the mutual information-based Learn++.NSE ensembles (designated LPP-MI). Here, the mutual information between the latest data set recorded from a certain participant and all other data sets from all the participants were computed. Latest data set and the two data sets with the most mutual information were used for the training of LPP-MI for that specific participant. Specifically, the data sets were added to the Learn ++ algorithm in the following order: (1) the set with second most mutual information, (2) the set with most mutual information, and (3) most recent data set. The second group contained the standard Learn++.NSE ensembles (denoted as LPP-S). For each participant, LPP-S was formed using the three training data sets corresponding to that specific participant. An LDA classifier was also trained for each participant on their last session's calibration phase in order to compare performance under typical calibration procedures. The three classifiers were then compared by examining their accuracies over the test data which was not used for training.

4. RESULTS AND DISCUSSION

Figure 2 displays the accuracies obtained by the LPP-MI and LPP-S ensemble as well as the LDA classifier. In 7 of 10 cases, the LPP-MS ensemble outperformed the LPP-S one. In two cases, they had very similar performance. In only one case did the standard Learn++.NSE ensemble accuracy exceeded the mutual information ensemble. In 9 of 10 cases, the LPP-MI ensemble outperformed the standard LDA classifier. It should be noted that the case where LDA outperformed LPP-MI is different than the case where the LPP-S ensemble outperformed the LPP-MI ensemble.

All the accuracies were used in a Wilcoxon signed rank test [14]. A p-value of .0039 was reported, indicating that there is a significant difference between the accuracies obtained by the LPP-MI and LPP-S ensembles. A p-value of .006 was calculated when comparing the LPP-MI ensembles and the LDA classifiers, demonstrating that LPP-MI also outperforms the standard LDA classifier.

The incorporation of a mutual information layer over Learn++.NSE shows a significant improvement in performance over limiting candidate data sets to those of the current user. The greatest accuracy increases were seen in data sets that traditionally performed poorly. One interpretation of this result is that the previous classifiers in the traditional Learn++.NSE ensemble were poorly trained or that the nonstationarity between data sets caused them to be detrimental in a smaller ensemble size. The mutual information step appears to have identified the data sets most like the incoming data and populated the ensemble with data sets that would improve accuracy.

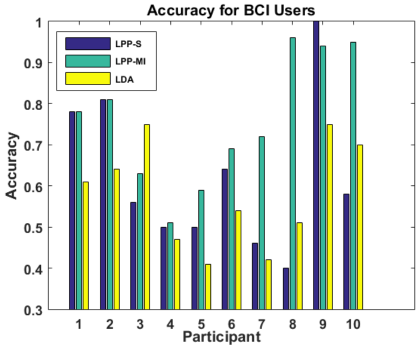

Fig. 2: Accuracy results comparing the LPP-MI and LPP-S Ensembles

There was one data set where the LPP-MI ensemble was outperformed by the LPP-S ensemble. This was the case for participant 9, where the LPP-S ensemble achieved perfect accuracy. For this participant, any other learning algorithm is likely to achieve worse results, but even in this case the LPP-MI ensemble achieved an accuracy of 94%. It should also be noted that the LDA classifier outperformed both ensembles for participant 3's data set. One possible explanation for this could be that there is so much nonstationarity among different sessions that the LDA trained using the training data of the last session performs the best over test data of the same session.

5. CONCLUSION AND FUTURE WORK

The use of mutual information in data set selection holds promise for BCI systems that have access to training data from multiple users. Mutual information allows the system to determine the best data sets for classification. The use of an ensemble classification technique allows a multi-user BCI to best leverage the abundance of data available.

The main benefits to this approach over other methods is the ability to apply transfer toward SSVEP systems while using fewer computational resources. Learn++.NSE is not computationally intensive, and the estimation of the augmented covariance matrix for the mutual information is feasible. Since SSVEP systems are designed to be simple, the transfer learning algorithm should reflect this.

This type of approach has room for further development. Modifications of the Learn++.NSE algorithm can be considered. An alternative way of calculating voting weights might include the mutual information instead of relying on past classification error.

6. REFERENCES

[1] Murat Akcakaya, Betts Peters, Mohammad Moghadamfalahi, Student Member, Aimee R Mooney, Umut Orhan, Student Member, Barry Oken, Deniz Erdogmus, Senior Member, and Melanie Fried-oken, "Noninvasive Brain – Computer Interfaces for Augmentative and Alternative Communication," *IEEE Reviews in Biomedical Engineering*, vol. 7, pp. 31–49, 2014.

[2] Yijun Wang and Tzyy-ping Jung, "A Collaborative Brain-Computer Interface for Improving Human Performance," *PLOS ONE*, vol. 6, no. 5, 2011.

[3] Yijun Wang, Yu-te Wang, and Tzyy-ping Jung, "A Collaborative Brain-Computer Interface," in *20111 4th International Conference on Biomedical Engineering and Informatics*, 2011, pp. 583–586.

[4] Mark Wronkiewicz, Eric Larson, Adrian K C Lee, Timothy Zeyl, Erwei Yin, and Michelle Keightley, "Integrating Dynamic Stopping, Transfer Learning, and Language Models in an Adaptive Zero-Training ERP Speller," *Journal of Neural Engineering*, 2014.

[5] Setare Amiri, Ahmed Rabbi, Leila Azinfar, and Reza Fazel-Rezai, "A Review of P300, SSVEP, and Hybrid P300/SSVEP Brain- Computer Interface Systems," *In-Tech Open*, pp. 195–213, 2013.

[6] Chuan Jia, Xiaorong Gao, Bo Hong, and Shangkai Gao, "Frequency and Phase Mixed Coding in SSVEP-Based Brain – Computer Interface," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 1, pp. 200–206, 2011.

[7] Matt Higger, Murat Akcakaya, Hooman Nezamfar, Gerald Lamountain, Umut Orhan, and Deniz Erdogmus, "A Bayesian Framework for Intent Detection and Stimulation Selection in SSVEP BCIs," *IEEE Signal Processing Letters*, vol. 22, no. 6, pp. 743–747, 2015.

[8] Masaki Nakanishi, Student Member, Yijun Wang, Yu-te Wang, Student Member, Yasue Mitsukura, Tzyy-ping Jung, and Senior Member, "Enhancing Unsupervised Canonical Correlation Analysis-Based Frequency Detection of SSVEPs by Incorporating Background EEG," in *36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014, pp. 3053–3056.

[9] Ryan Elwell, Robi Polikar, and Senior Member, "Incremental Learning of Concept Drift in Nonstationary Environments," *IEEE Transactions on Neural Networks*, vol. 22, no. 10, pp. 1517–1531, 2011.

[10] Lars Schmidt-thieme and Frank Hutter, "Beyond Manual Tuning of Hyperparameters," *Kunstliche Intelligen*, vol. 29, no. 4, pp. 329–337, 2015.

[11] Marcelo S Alencar, *Information Theory*, Momentum Press, New York, 2015.

[12] Ivan W Selesnick, L Markus, and C Sidney Bums, "Filters without Specified Transition Bands," *Proceedings of the 1995 International Conference on Acoustics, Speech, and Signal Processing*, vol. 44, no. 8, pp. 1879–1892, 1996.

[13] Peter D. Welch, "The Use of Fast Fourier Transform for the Estimation of Power SPectra: A Method Based on Time Averaging over Short, Modified Periodograms," *IEE Transaction on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.

[14] Frank Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.