Proceedings of the
7th Graz Brain-Computer Interface Conference 2017 DOI: 10.3217/978-3-85125-533-1-28

OPTIMAL TRANSPORT APPLIED TO TRANSFER LEARNING FOR P300 DETECTION

N.T.H. Gayraud[1], A. Rakotomamonjy[2], M. Clerc[1]

[1]Université Côte d'Azur, Inria, France
[2]Rouen University, France

E-mail: nathalie.gayraud@inria.fr

ABSTRACT: Brain Computer Interfaces suffer from considerable cross-session and cross-subject variability, which makes it hard for classification methods to generalize. We introduce a transfer learning method based on regularized discrete optimal transport with class labels in the interest of enhancing the generalization capacity of state-of-the-art classification methods. We demonstrate the potential of this approach by applying it to offline cross-subject transfer learning for the P300-Speller paradigm. We also simulate an online experiment to assess the feasibility of our method. Results show that our method is comparable to -and sometimes even outperforms- session-dependent classification.

INTRODUCTION

Brain Computer Interfaces (BCI) are a means of communication that connect a human brain and a machine, bypassing any other neurological output. In particular, during a non-invasive EEG-based BCI session, neurophysiological signals are acquired, processed, and transformed into commands, for which the user receives some form of feedback. Due to the very low Signal to Noise Ratio (SNR) non-invasive EEG-based BCI suffer from, advanced signal processing and machine learning techniques need to be employed for the intermediate steps [1].

EEG signals also suffer from a high amount of session-to-session and subject-to-subject variability, whose sources are diverse [2]. It can be due to the use of different acquisition means, to varying conditions during the day of the acquisition, to neurophysiological differences between one user and another, or to the fact that mental states and levels of concentration change from one session to another. Therefore, the classifier used to label mental tasks needs to be trained before every use, a task commonly referred to as calibration. Furthermore, because variability can occur within a session, the BCI may need to be recalibrated during its use. Calibration can last several minutes depending on the subject; it lists high among the reasons why the use of BCI is still not widespread.

The design of a robust transfer learning classification algorithm has been a subject of broad interest in the BCI community. The first attempts towards zero-training BCI are made for the Motor Imagery paradigm. Some of these methods rely on recovering spatial filters to project the samples onto a space where a pre-trained classifier will generalize well [3, 4], others on the use of adaptive or ensemble methods [5, 6]. The latter are also used in cross-session and cross-subject classification for P300-based BCI [7, 8], along with approaches under the Riemannian framework [9].

This work handles transfer learning classification by treating cross-session and cross-subject variability as a unsupervised domain adaptation problem. Recent works by Courty et al. [10] propose a solution based on regularized optimal transport to tackle the problem of classifying unlabeled test data that belong to a different domain from which the training data is drawn. Transportation theory applications to BCI have been researched under a mostly theoretical framework in the works of Ma et al. [11] towards generalizing the Posterior Matching Scheme to arbitrarily many dimensions.

Our contribution is a methodological framework based on regularized optimal transport with class labels which can be used for transfer learning alongside existing classifiers. In this paper, it is assessed through offline cross-subject experiments under the P300-Speller paradigm.

In the following sections, we first describe the problem formally and introduce notations. We proceed by describing our method, the dataset used in the experiments, and the experiments themselves. Then, we present our results and discuss them. Finally, we give our conclusions and propose future extensions.

MATERIALS AND METHODS

Transfer learning as a domain adaptation problem

Let $\mathbf{S} = \{(x_i, y_i)\}_{i=1}^{N}$ be the set of data acquired during a BCI session, that is, the set of N extracted feature vectors $\mathbf{X} = \{x\}_{i=1}^{N} \subset \mathbb{R}^d$ of dimension d coupled with the corresponding labels $\mathbf{Y} = \{y\}_{i=1}^{N}$. Furthermore, let $\mathbf{P}(x) \in \mathcal{P}(\Omega)$ denote the probability distribution from which the samples in $\mathbf{X}$ are drawn, where $\Omega \in \mathbb{R}^d$ is a measurable space of dimension d and $\mathcal{P}(\Omega)$ the set of all probability measures over the domain Ω.

We denote by $\mathbf{S}^e$ an existing session for which the labels are available, and by $\mathbf{S}^n$ a new session for which they are unknown. We seek to train a classifier to recover the unknown labels $\mathbf{Y}^n$. However, as a result of cross-session and cross-subject variability, most classifiers do not give accurate results about $\mathbf{Y}^n$ when trained on $\mathbf{S}^e$. This effect can be modeled as a domain adaptation problem, known as covariate shift [2]: while the conditional probability distributions $\mathbf{P}(y|x^e)$ and $\mathbf{P}(y|x^n)$ are equal, the same does not hold for the probability distributions of $\mathbf{P}(x^e) \in \mathcal{P}(\Omega^e)$ and $\mathbf{P}(x^n) \in \mathcal{P}(\Omega^n)$. Assuming that a transformation causes the drift between domains Ω_n and Ω_e, we propose to recover a transport plan to map the new features onto the domain of the existing features (Ω_e) using Optimal Transportation (OT) theory.

Regularized discrete OT with class labels

OT theory studies a problem known as the Monge-Kantorovic transportation problem [12]. This problem can be intuitively understood as the search for the optimal way to transport mass between two probability distributions. The optimization criterion is the minimization of a transportation cost; typically, the cost function represents some metric between the random variables of each distribution. Also, constraints may be imposed so that the mass is preserved during the transport. Since we only have a fixed number of samples from each set, the discrete adaptation of the OT problem boils down to matching empirical measures μ_e, μ_n of $\mathbf{P}(x^e)$ and $\mathbf{P}(x^n)$.

We can now formally define regularized discrete OT with class labels in the following way: consider the estimated empirical marginal distributions $\mu_e = \sum_{i=1}^{N_e} p_i \delta_{x_i^e}$ and $\mu_n = \sum_{i=1}^{N_n} p_i \delta_{x_i^n}$ of the samples in $\{x_i^e\}_{i=1}^{N_e} = \mathbf{X}^e$ and $\{x_i^n\}_{i=1}^{N_n} = \mathbf{X}^n$, where δ_{x_i} is the Dirac function at $x_i \in \mathbb{R}^d$ and p_i is the probability mass associated to the i^{th} sample, $\sum_{i=1}^{N} p_i = 1$. We look for a probabilistic coupling $\gamma_0 \in \mathcal{B}$ satisfying the following minimization problem:

$$\gamma_0 = \underset{\gamma \in \mathcal{B}}{\operatorname{argmin}} \langle \gamma, \mathbf{C} \rangle_{\mathbf{F}} + \lambda \mathbf{R}_s(\gamma) + \eta \mathbf{R}_c(\gamma) \tag{1}$$

where $\langle \cdot \rangle_{\mathbf{F}}$ is the Frobenius dot product, and $\mathcal{B}$ is the set of all probabilistic couplings between μ_e and μ_n, $\mathcal{B} = \left\{ \gamma \in (\mathbb{R}^+)^{N_e \times N_n} \mid \gamma \mathbf{1}_{N_n} = m_e, \gamma^{\mathbf{T}} \mathbf{1}_{N_e} = m_n \right\}$ where $\mathbf{1}_d$ denotes a d-dimensional vector of ones and $m \in \mathbb{R}^N$ denotes a vector of probabilities, each probability associated to a point in feature set $\mathbf{X}$.

The first term of equation 1 is the discrete adaptation of the Kantorovic formulation of the OT problem [13]. $\mathbf{C}$ is the cost function matrix, whose elements correspond to a metric between two points, $\mathbf{c}_{ij} = d(x_i^e, x_j^n)$, $x_i^e \in \mathbf{X}^e, x_j^n \in \mathbf{X}^n$; it can be intuitively understood as the effort required to move a probability mass from x_i^e to x_j^n. In this work, the metric we use is the squared Euclidean distance $d(x_i^e, x_j^n) = \|x_i^e - x_j^n\|_2^2$, as it guarantees the existence of a unique coupling [12]. When the squared euclidean distance is used as the cost function, the first term leads to a sparse version of γ_0

The second term regularizes γ_0 by its entropy [14]:

$$\mathbf{R}_s(\gamma) = \lambda \sum_{i,j} \gamma(i,j) \log \gamma(i,j) \tag{2}$$

This allows for smoother variants of γ_0, whose sparsity gradually decreases as λ increases, and renders the transport more robust to noise. Moreover, $\mathbf{R}_s(\gamma)$ can also be interpreted as a Kullback-Leibler divergence between γ and a uniform joint probability $\gamma_u = \frac{1}{N_e N_n}$, which allows for the use of a computationally efficient algorithm based on Sinkhorn-Knopp's scaling matrix approach [15].

The third term is a regularizer, proposed by Courty et al. [10], based on group sparsity which makes use of the available class labels of session $\mathbf{S}^e$:

$$\mathbf{R}_c(\gamma) = \sum_j \sum_{cl} \|\gamma(\mathcal{I}_{cl}, j)\|_2 \tag{3}$$

where $\mathcal{I}_{cl}$ denotes the set of indices belonging to class $cl \in \{Target, Nontarget\}$. In this way, the j-th element $x_j^n \in \mathbf{X}^n$ will not be coupled with elements from $\mathbf{X}^e$ that belong to different classes.

OT applied to P300-based BCI

Based on the previous formulation of the OT problem, we propose a transfer-learning method whose three main steps are (a) feature extraction, (b) transportation of the new features to the domain of the existing set and (c) label prediction. The pipeline of our method is illustrated in Fig. 1.

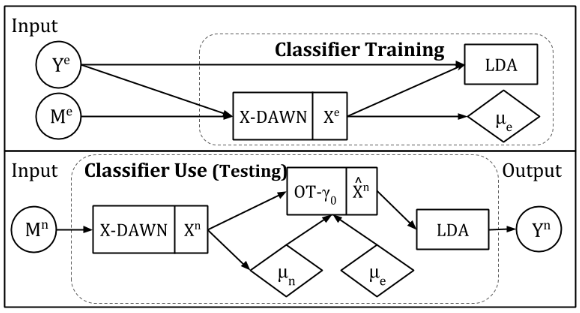


Figure 1: Pipeline of the method. During the training process, an existing set $\mathbf{M}^e$ is given as input along with the corresponding labels $\mathbf{Y}^e$. Then, (a) the X-DAWN spatial filters are learned and (b) the extracted features $\mathbf{X}^e$ are used to estimate μ_e and train the LDA classifier. When a new set $\mathbf{M}^n$ is given as input to the trained classifier, (a) the trained X-DAWN filters are used to extract features $\mathbf{X}^n$, (b) μ_n and γ_0 are estimated, and (c) $\hat{\mathbf{X}}^n$ is computed and given as input to the LDA classifier, which estimates $\mathbf{Y}^n$.

Let $m(t) \in \mathbb{R}^C$ be a measurement extracted from a downsampled EEG signal over C electrodes at time t during a P300-Speller session. After pre-processing, $M_i \in \mathbb{R}^{C \times T}$ denotes the i^{th} trial whose columns are T consecutive measurements. From $\{M_i\}_{i=1}^{N_e} = \mathbf{M}^e$ and the corresponding labels $\mathbf{Y}^e$, we learn spatial filters using the X-DAWN algorithm [16], and project each M_i onto the first N_f X-DAWN filters, yielding feature vectors $x_i \in \mathbb{R}^{N_f \times T}$.

We proceed by computing γ_0 according to equation (1) and use it to map $\mathbf{X}^n$ onto $\mathbf{\Omega}^e$ by computing a transformation based on barycentric mapping [10],

$$\hat{\mathbf{X}}^n = \operatorname{diag}(\gamma_0^\top \mathbf{1}_{N_e})^{-1} \gamma_0^\top \mathbf{X}_e \tag{4}$$

Each $x^n \in \mathbf{X}^n$ will thus be mapped onto the weighted barycenter of the features of $\mathbf{X}^e$ that it was coupled with.

In the end, a Linear Discriminant Analysis (LDA) classifier is trained on $\mathbf{S}^e$, and used to predict the labels $\{y_i^n\}_{i=1}^{N_n} = \mathbf{Y}^n$ that correspond to $\{\hat{\mathrm{x}}_i^n\}_{i=1}^{N_n} = \hat{\mathbf{X}}^n$.

Dataset Description

The first dataset used in our experiments, Dataset A, consists of EEG signals recorded during P300-Speller sessions that were conducted by adult patients suffering from Amyotrophic Lateral Sclerosis. The experiment took place in the premises of the Nice University hospital, and had been approved by the local ethics committee CPP Sud Méditerrannée [17]. Each subject participated in three free-spelling sessions, each one preceded by a calibration session. In this paper, we use the calibration sessions of 12 randomly selected patients. In each session there are in total 200 trials in the $Target$ class (considered to contain the elicited P300 component) and 1000 trials in the $Nontarget$ class.

Dataset B includes EEG signals from four healthy subjects, which were recorded during P300-speller sessions conducted in the premises of Inria Sophia-Antipolis Mediterranée. Each subject participated in two free-spelling sessions, each one preceded by a calibration session. Again, we only include the calibration sessions, each one containing 66 trials in the $Target$ class and 330 trials in the $Nontarget$ class.

In both datasets, a Refa-8 amplifier (ANT) with 12 electrodes (Fz,C3,Cz,C4,P7,P3,Pz,P4,P8,O1,Oz,O2) was used for the recording. The EEG signals are filtered with a 5th order Butterworth filter between 1 and 15Hz. Each signal is then downsampled from 256 Hz to 64Hz and separated into trials $M_i \in \mathbb{R}^{C \times T}$, where $C = 12$ and $T = 32$ to account for a 0.5s epoch starting at the time of the flash.

Cross-subject experiments

To demonstrate the potential of our approach, we initially conduct two offline experiments using Dataset A, the difference between them lying in the composition of the training (existing) set. In both cases, the labels associated to the testing (new) set are not taken into consideration during the experiment, and are only used for evaluation purposes.

In the first experiment we assess the generalization capacity of our classifier in *pairwise transfer learning* experiments. For each experiment, the training set consists of a single session, that is, a set $\mathbf{M}_i^e$ of trials along with the corresponding labels in $\mathbf{Y}_i^e$ and the test set is $\mathbf{M}_j^n$, where $i, j \in I = \{A1, A2, ..., A12\}$ denote the subject index, and $i \neq j$. The cardinality of each set is $N_e = N_n = 1200$.

For the second experiment, we evaluate the performance of our classification method when trained with a larger training set by performing *Leave-One-Out transfer learning*. As the test session contains data from a single session, $\mathbf{X}_{j \in I}^n$, here, the training set consists of the entire dataset but session j. Hence, $\mathbf{M}^e = \bigcup_{i \in I_j^-} \mathbf{M}_i$ and $\mathbf{Y}^e = \bigcup_{i \in I_j^-} \mathbf{Y}_i$, where $I_j^- = I - \{j\}$ denotes the set of indices of all subjects except subject j. In this setting, $N_e = 13200$ and $N_n = 1200$.

Since the size of the training set is prohibitively large to allow for the fast computation of γ_0, we use an ensemble classifier method known as Bootstrap Aggregating (BA) or Bagging. Introduced by Breiman in 1996 [18], BA has often been used in BCI [19, 20] to enhance classification results. The pipeline of our method combined to BA is illustrated in Fig. 2. Initially, BA creates k subsets of length l, called bootstraps, by sampling the training set uniformly and with replacement. We train an instance of our classifier for each bootstrap. During testing, each instance produces a prediction; all of the predictions are aggregated via a voting scheme, that is, a majority vote, to produce the final result.

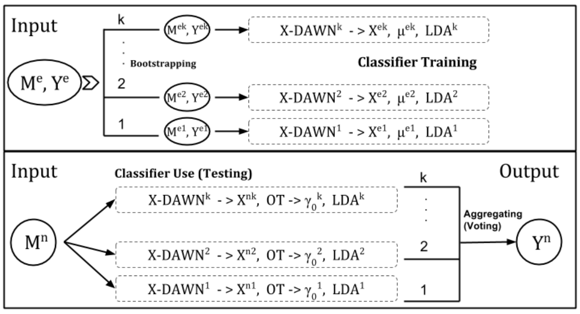

Input
k
Mek, Yek
X-DAWNk - > Xek, μek, LDAk
Bootstrapping
Classifier Training
Me, Ye
2
Me2, Ye2
X-DAWN2 - > Xe2, μe2, LDA2
1
Me1, Ye1
X-DAWN1 - > Xe1, μe1, LDA1
Input
Classifier Use (Testing)
k
Output
X-DAWNk - > Xnk, OT -> γ0k, LDAk
Mn
Aggregating (Voting)
Yn
X-DAWN2 - > Xn2, OT -> γ02, LDA2
2
X-DAWN1 - > Xn1, OT -> γ01, LDA1
1

Figure 2: Pipeline of our method with BA. Initially, BA creates k subsets from the training set. Then, an instance of our classifier is trained for each subset. During testing, the new set $\mathbf{M}^n$ is given as input to each instance. All instances produce a prediction, and all predictions are aggregated via voting to produce the final result.

Finally, we simulate one online experiment per session in Dataset B, using the pairwise transfer learning classifier from dataset A that produced the best performance. The simulation proceeds in the following way: every $N_F = 36$ trials, the feature vector set is extracted from $\{M_i\}_{i=1}^{N_F} = \mathbf{M}^n$, the transport map between $\mathbf{X}^e$ and $\{x_i\}_{i=1}^{N_F} = \mathbf{X}^n$ is computed, and $\{y_i\}_{i=1}^{N_F} = \mathbf{Y}^n$ is generated from the mapped set $\{\hat{x}_i\}_{i=1}^{N_F} = \hat{\mathbf{X}}^n$. Note that we keep the chronological ordering of the trials within each test session.

For all experiments, we use the first and last two X-DAWN filters during feature extraction, resulting in a total of $N_f \times T = 128$ extracted features. The best values for the OT regularization terms are searched and selected in $\lambda, \eta \in \{0.01, 0.1, 1, 10, 100\}$. For classifiers using the BA method, $k = 20$ bootstraps of length $l = 500$ were used, and we drew the same number of elements from each class to remedy the issue of class imbalance.

RESULTS

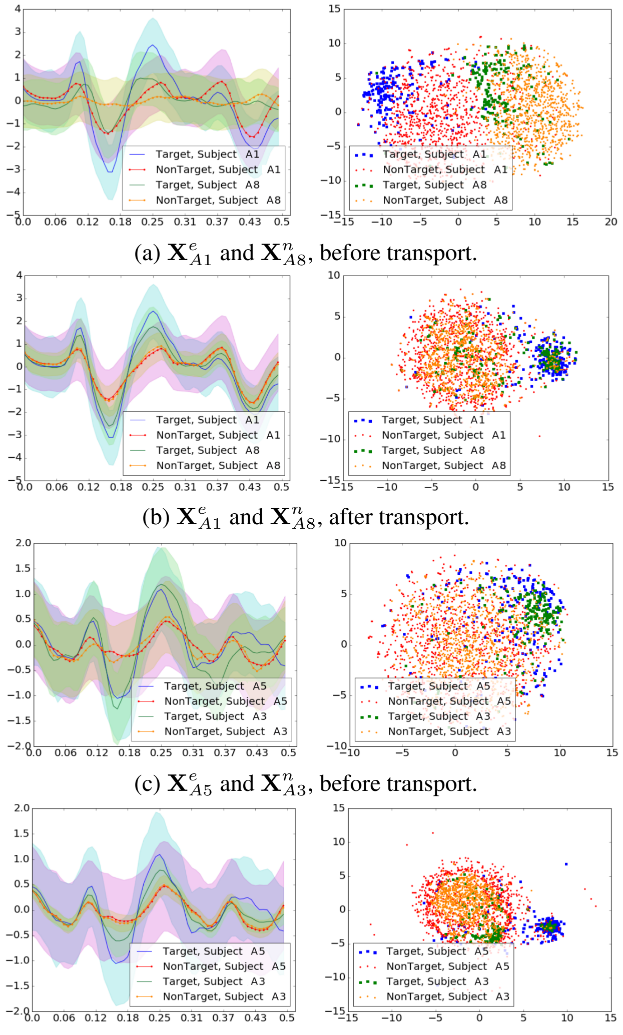
We introduce this section by illustrating an example of a transport between two pairs of sessions. Then, we report

Table 1: Pairwise transfer learning. Columns display the average AUC value and standard deviation over 11 experiments where the classifier is trained with the corresponding existing session; the last column is the average over 132 experiments. The first row shows the results obtained by XD+LDA, while the second row shows the results from XD+OT+LDA.

Existing Session	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	**Avg.**
XD+LDA	0.535 ± 0.05	0.562 ± 0.07	0.598 ± 0.09	0.600 ± 0.07	0.591 ± 0.10	0.595 ± 0.08	0.570 ± 0.06	0.578 ± 0.06	0.516 ± 0.02	0.566 ± 0.07	0.553 ± 0.06	0.526 ± 0.02	**0.566** ± **0.03**
XD+OT+LDA	0.627 ± 0.07	0.539 ± 0.02	0.567 ± 0.06	0.548 ± 0.04	0.611 ± 0.11	0.598 ± 0.07	0.560 ± 0.06	0.490 ± 0.17	0.518 ± 0.01	0.551 ± 0.04	0.583 ± 0.05	0.585 ± 0.06	**0.565** ± **0.04**

Table 2: Leave-One-Out transfer learning. Columns display the AUC score when the classifier is trained with all of dataset A except for the corresponding new session. The last column is the average and standard deviation over 11 experiments. The first two rows show results obtained whithout OT; in the first row, the BA method is not used either. The third row displays the results when both BA and OT are used. The last row shows the AUC score of the Session-Dependent (SD) classifier of each session in dataset A.

New Session	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	**Avg.**
XD+LDA	0.804	0.500	0.529	0.567	0.528	0.593	0.577	0.506	0.525	0.689	0.690	0.721	**0.602** ± **0.099**
BA+XD+LDA	0.778	0.535	0.720	0.808	0.623	0.698	0.739	0.596	0.521	0.696	0.673	0.809	**0.683** ± **0.098**
BA+XD+OT+LDA	0.779	0.529	0.835	0.790	0.732	0.541	0.802	0.608	0.673	0.740	0.655	0.809	**0.708** ± **0.106**
SD Classifier	0.724	0.593	0.709	0.713	0.648	0.624	0.692	0.658	0.548	0.694	0.702	0.781	**0.673** ± **0.063**



(a) $\mathbf{X}^e_{A1}$ and $\mathbf{X}^n_{A8}$, before transport.

(b) $\mathbf{X}^e_{A1}$ and $\mathbf{X}^n_{A8}$, after transport.

(c) $\mathbf{X}^e_{A5}$ and $\mathbf{X}^n_{A3}$, before transport.

(d) $\mathbf{X}^e_{A5}$ and $\mathbf{X}^n_{A3}$, after transport.

Figure 3: Examples of the barycentric mapping induced by γ_0 for pairs of sessions. On the left we see the average response and standard deviation of the first X-DAWN filter projection for the $Target$ and $Nontarget$ classes. On the right side, we see the 2D projection of the features, projected using t-SNE.

the experimental results of our method, which we refer to as XD+OT+LDA. Motivated by the high level of imbalance between the $Target$ and $Nontarget$ class, we use the Area Under the receiver operating characteristic Curve (AUC) as our performance metric.

We display two examples of the estimated optimal transport in Fig. 3. In the first, the training and testing feature vector sets are $\mathbf{X}^e_{A1}$ and $\mathbf{X}^n_{A8}$ respectively, while in the second, $\mathbf{X}^e_{A5}$ and $\mathbf{X}^n_{A3}$. Fig. 3a and 3c show the original datasets, while Fig. 3b and 3d illustrate the outcome after computing $\hat{\mathbf{X}}^n_{A8}$ and $\hat{\mathbf{X}}^n_{A3}$. On the right side, we display a 2D projection of the features using t-distributed stochastic neighbor embedding (t-SNE) [21]. On the left side, we can observe the average response and standard deviation of the first X-DAWN filter, estimated on $\mathbf{X}^e_{A1(A5)}$, for both sessions and both classes. By looking closely at Fig. 3b and 3d, we can see that the transport causes a decrease in the variance of the response, for both the $Target$ and $Nontarget$ classes.

Pairwise Transfer Learning

The results of pairwise transfer learning can be seen on Tab. 1. To evaluate the performance of our method, we compare it to the performance of an XD+LDA classifier, i.e. an LDA classifier and X-DAWN features trained on $\mathbf{M}^e_{i\in I}$, $\mathbf{Y}^e_{i\in I}$, where no transport takes place. For each training session, we display the average AUC score and the standard deviation of the 11 experiments conducted with its corresponding classifier, where each one of the remaining session was used as the test session, $\mathbf{M}^n_{j\in I}$, $j \neq i$. At first glance, the two methods seem to perform equally well, yielding an average score of ~ 0.56. We note however that the best performance overall is the one of our method when trained with session S^e_{A1}, which is equal to 0.627.

Leave-One-Out Transfer Learning

On Tab. 2, we display the results of Leave-One-Out transfer learning; for each test session $\mathbf{M}^n_{j\in I}$, we trained a BA+XD+OT+LDA classifier (a combination of the BA ensemble method and our method, where each bootstrap is used to train an XD+OT+LDA classifier), using the union of the remaining sessions. For comparison purposes, we show the corresponding results of two additional transfer learning classifiers: a classifier without OT, in which BA is used to enhance the performance of an XD+LDA classifier, and those of an XD+LDA classifier, where neither BA nor OT are used. Finally, the performance of the session-dependent (SD) LDA classifier trained on X-DAWN features, computed after 5-fold cross-validation, is also displayed at the bottom.

Our findings demonstrate that merely using the BA method produces better results than the simple XD+LDA classifier. The average score of the BA+XD+LDA classification method is equal to 0.683, while the average performance of the session-dependent classifiers is 0.673. On top of that, when we use OT, we boost the performance even more producing an average performance equal to 0.708.

Online Simulation

After obtaining the results from pairwise transfer learning and observing that the XD+OT+LDA classifier trained with $\mathbf{M}^e_{A1}$,$\mathbf{Y}^e_{A1}$ generated the best performance, we used it to simulate an online experiment using each one of the four sessions $\{B1, B2, B3, B4\}$ in Dataset B as the test session (Sim 1). A label set $\mathbf{Y}^{n_i}$ was produced every $N_F = N_n = 36$ trials; since each test set contains 396 trials, a total of 11 label sets $\mathbf{Y}^{n_i}$, $i \in 1, \cdots, 11$, were generated in the course of each experiment. For each simulation, we report on Tab. 3 the average AUC and standard deviation over all label sets $\mathbf{Y}^{n_i}$. We compare it to an analogous online simulation using the XD+LDA classifier trained with $\mathbf{M}^e_{A4}$,$\mathbf{Y}^e_{A4}$, which generated the best score in pairwise transfer leaning among all XD+LDA classifiers (Sim 2). The AUC scores for the session specific classifiers of each test session, computed after 5-fold cross-validation, are also reported.

Table 3. Results from the online simulations. Sim 1 is the simulation where the XD+OT+LDA classifier is trained with Dataset A session $A1$, while Sim 2 is the simulation where the XD+LDA classifier is trained with Dataset A session $A4$.

Test Sess.	B1	B2	B3	B4	**Avg.**
Sim 2	0.66 ± 0.11	0.60 ± 0.07	0.76 ± 0.14	0.86 ± 0.09	**0.72 ± 0.11**
Sim 1	0.78 ± 0.12	0.71± 0.14	0.77 ± 0.10	0.69 ± 0.10	**0.74 ± 0.04**
SD cl.	0.78	0.80	0.68	0.85	**0.77**

We can see that, 3 times out of 4, our best pairwise transfer learning classifier outperforms the best pairwise transfer learning XD+LDA classifier. For subject P3, both classifiers score better than the session-dependent classifier.

Computation time

Regarding the complexity of our method, the average computation time needed to compute each transport map in pairwise transfer learning is equal to ∼8 sec, compared to ∼35 sec for Leave-One-Out transfer learning. Since the test sets are much smaller in the online simulation, the average computation time is ∼0.86 seconds per test set. All experiments were conducted on a computer with a 2.8 GHz Intel i7 processor and 8 GB of RAM.

DISCUSSION

The results presented in the previous section are strong indicators that the OT approach can effectively enhance transfer learning.

Regarding the mapping itself, the examples illustrated on Fig. 3 give us some insight on the process and how it acts on the components of the EEG signal. We see that, after the transport, the average values of the first X-DAWN component of $Target$ and $Nontarget$ class match quite well, especially for the $Nontarget$ class. However, due to the presence of a much larger number of $Nontarget$ class elements in the training set, it appears that samples whose elicited P300 component is weak are drawn to the training $Nontarget$ class barycenter. Our decision to select an equal number of elements in each class to generate the bootstraps for the BA method finds its motivation in this observation.

Another product of barycentric mapping is the observed decrease in the variance of the responses of the X-DAWN filters, seen on Fig. 3b and 3d. This is a consequence of the choices for parameter η and λ. For high values, the "new" data points tend to be drawn to the mean of each class in the existing set. Lower parameters generate a larger variance in each mapped class; however, they also reduce the separability of the classes in the mapped feature vector set.

Despite the fact that pairwise transfer learning did not produce conclusive result in favor of OT, our method generated the two highest AUC scores. Concerning Leave-One-Out transfer learning, we remark that the high level of variability in the training set, due to the fact that it contains trials from many different subjects, actually affects the average prediction accuracy in a positive way. Moreover, the use of the BA method induces a general improvement in prediction accuracy, and leads to even better results when OT-based mapping is used. However, it increases the computational time, since the computational cost of computing γ_0 depends on the size of the number of elements in each set.

Fortunately, the computational time of ≤ 1 sec for each small 36-trial set in the online simulation is low enough to allow for a fast online implementation. Our findings during online simulations show that our method outperforms the state-of-the-art classification method. These observations encourage us to continue our research towards the implementation of a zero-training OT-based classifier.

CONCLUSION

In this work, we have demonstrated that Discrete Regularized OT can be used in cross-subject transfer learning to improve the generalization capacities of existing P300-based classification methods. The results obtained by OT-based classifiers indicate that our method has the potential to cancel the need for calibration.

Nevertheless, we are most interested in understanding why some sets seem to contain more information than others. In future works, we would investigate which are the characteristics that qualify a good "map-to" candi-

date. Subsequently, instead of using one specific session, or a number of bootstraps generated from specific sessions, we would be using a number of prototypical training sets that carry these characteristics. In that case, the voting scheme could be bypassed by session-dependent selection of one of these subsets with respect to a metric, such as the Kullback-Leibler divergence or the Information Geometry derived Riemannian distance.
While in this paper we concentrate on cross-subject transfer learning, this work can be extended to cross-session transfer learning or even be used to improve classification results within a session classifier. Finally, we are also interested in using this approach under Motor Imagery BCI paradigms.

ACKNOWLEDGEMENTS

The Nice University Hospital is acknowledged for providing the dataset used in this paper. We would also like to thank Rémi Flamary for letting us use his toolbox, Python Optimal Transport (POT).

REFERENCES

[1] Clerc M, Bougrain L, Lotte F. Brain-Computer Interfaces 1, Part 2, Wiley-ISTE (2016).

[2] Clerc M, Daucé E, Mattout J. Adaptive Methods in Machine Learning. In: Clerc M, Bougrain L, and Lotte F. Brain-Computer Interfaces 1. Wiley-ISTE, 2016, pp. 209-232.

[3] Krauledat M, Tangermann M, Blankertz B, Müller KR. Towards zero training for brain-computer interfacing. PloS one. 2008;3(8):e2967.

[4] Reuderink B, Farquhar J, Poel M, Nijholt A. A subject-independent brain-computer interface based on smoothed, second-order baselining, in Proc. IEEE EMBC, 2011, Boston, MA, USA, 2011, 4600-4604.

[5] Blankertz B, Dornhege G, Krauledat M, Müller KR, Curio G. The non-invasive Berlin brain–computer interface: fast acquisition of effective performance in untrained subjects. NeuroImage. 2007;37(2):539-550.

[6] Fazli S, Popescu M, Danóczy M, Blankertz B, Müller KR, Grozea C. Subject-independent mental state classification in single trials. Neural networks. 2009;22(9):1305-1312.

[7] Rakotomamonjy A, Guigue V. BCI competition III: dataset II-ensemble of SVMs for BCI P300 speller. IEEE transactions on biomedical engineering. 2008;55(3):1147-1154.

[8] Lu S, Guan C, Zhang H. Unsupervised brain computer interface based on intersubject information and online adaptation. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2009;17(2):135-145.

[9] Barachant A, Congedo M. A plug&play p300 bci using information geometry. arXiv preprint. 2004;arXiv:1409.0107.

[10] Courty N, Flamary R, Tuia D, Rakotomamonjy A. Optimal transport for Domain adaptation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2016.

[11] Ma R, Coleman TP. Generalizing the posterior matching scheme to higher dimensions via optimal transportation, in Proc. 49th Annual Allerton Conference on Communication, Control, and Computing , IEEE, 2011 , Allerton, IL, USA 96-102.

[12] Villani C. Optimal transport: old and new, Springer Science & Business Media, (2008).

[13] Kantorovitch L. On the translocation of masses. Management Science. 1958;5(1):1-4.

[14] Cuturi M. Sinkhorn distances: Lightspeed computation of optimal transport, in Proc. Advances in Neural Information Processing Systems, 2013, 2292-2300.

[15] Knight PA. The Sinkhorn–Knopp algorithm: convergence and applications. SIAM Journal on Matrix Analysis and Applications. 2008;30(1):261-275.

[16] Rivet B, Souloumiac A, Attina V, Gibert G. xDAWN algorithm to enhance evoked potentials: application to brain–computer interface. IEEE Transactions on Biomedical Engineering. 2009;56(8):2035-2043.

[17] Clerc M, Mattout J, Maby E, Devlaminck D, Papadopoulo T, Guy V, Desnuelle C. Verbal communication through brain computer interfaces, in Proc. Interspeech-14th Annual Conference of the International Speech Communication Association, 2013.

[18] Breiman L. Bagging predictors. Machine learning. 1996;24(2):123-140.

[19] Sun S, Zhang C, Zhang D. An experimental evaluation of ensemble methods for EEG signal classification. Pattern Recognition Letters. 2007;28(15):2157-2163.

[20] Blankertz B, Dornhege G, Müller KR, Schalk G, Krusienski D, Wolpaw JR, et al. Results of the BCI Competition III, in Proc. BCI Meeting, 2005.

[21] van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research. 2008;9(Nov):2579-2605.