Domain adaptation with optimal transport improves EEG sleep stage classifiers

Stanislas Chambon, Mathieu N. Galtier, Alexandre Gramfort

Domain adaptation with optimal transport improves EEG sleep stage classifiers

Stanislas Chambon*, Mathieu, N. Galtier*,
*Research & Algorithms Team, Rythm
Paris, France
stanislas@rythm.co

Alexandre Gramfort†
†Inria, Université Paris-Saclay,
Palaiseau, France
alexandre.gramfort@inria.fr

***Abstract*—Low sample size and the absence of labels on certain data limits the performances of predictive algorithms. To overcome this problem, it is sometimes possible to learn a model on a large labeled auxiliary dataset. Yet, this assumes that the two datasets exhibit similar statistical properties which is rarely the case in practice: there is a discrepancy between the large dataset, called the source, and the dataset of interest, called the target. Improving the prediction performance on the target domain by reducing the distribution discrepancy, between the source and the target domains, is known as Domain Adaptation (DA). Presently, Optimal transport DA (OTDA) methods yield state-of-the-art performances on several DA problems. In this paper, we consider the problem of sleep stage classification, and use OTDA to improve the performances of a convolutional neural network. We use features learnt from the electroencephalogram (EEG) and the electrooculogram (EOG) signals. Our results demonstrate that the method significantly improves the network predictions on the target data.**

***Index Terms*—EEG, sleep stage classification, domain adaptation, neural network, optimal transport**

I. Introduction

Training a supervised machine learning algorithm requires labeled data which might be limited or even absent in some cases. Using an auxiliary dataset, largely labeled, *a.k.a source domain*, enables to cope with this issue by training a predictive model that can then be used on a *target domain*. Yet, seldom are the source and target domains exactly identical in a statistical way: there is a domain discrepancy. *Domain Adaptation* (DA) methods [1] have been proposed to cope with this problem. The rationale behind DA is to find/adapt a feature representation that minimizes this discrepancy [2].

DA methods can be divided into two categories. Unsupervised DA methods perform adaptation without using labels in the target domain, whereas semi-supervised DA methods use labels from source domain as well as from a few labeled target samples. Furthermore, DA methods can be separated into two types: deep DA methods which aim at relearning the feature representation [3], [4] while shallow DA methods aim at reducing the domain discrepancy by applying an operation, typically linear, on a fixed feature representation [5]–[9].

Among shallow DA methods, optimal transport methods have exhibited promising performances on a wide range of problem [10], [11], in particular on some EEG signals [12]. The principle underlying these OTDA methods is finding a mapping that maps source samples into the convex hull of target samples such that the probability distribution of the mapped samples is similar to the target probability distribution. These proposed OTDA methods build on recent progress on computational optimal transport, such as entropic regularization which leads to a fast Sinkhorn-Knopp's algorithm [13].

Sleep stage classification, *a.k.a. sleep scoring*, is of great interest for understanding sleep and its disorders. Indeed, the sequence of sleep stages over a night is often used in the diagnosis of sleep disorders such as narcolepsy [14]. Traditionally, this exam is performed as follows. First a subject sleeps with a medical device which performs a polysomnography (PSG). It consists of electroencephalography (EEG) signals at different locations over the head, electrooculography (EOG) signals, electromyography (EMG) signals, and eventually more. Second, a human sleep expert looks at the different time series recorded over the night, and assigns to each 30 s time segment a sleep stage following a reference nomenclature such as American Academy of Sleep Medicine (AASM) rules [15]. The AASM rules identify 5 stages: Wake (W), Rapid Eye Movements (REM), Non REM1 (N1), Non REM2 (N2) and Non REM3 (N3) also known as deep sleep. They are characterized by distinct time and frequency patterns. They also differ in proportions over a night. Sleep scoring is a tedious and time consuming task which is furthermore subject to the scorer subjectivity and variability [14], yet it gives access to clinically relevant information about a patient.

From a machine learning perspective, the problem of automatic sleep scoring is an imbalanced multi-class classification problem. Several deep learning approaches have been proposed to learn an appropriate feature representation and classify sleep stages based on raw PSG signals or a time-frequency representation of these signals [14], [16], [17]. Despite leading to state-of-the-art performances on various datasets, training a deep architecture requires a large quantity of data [16] which might not be available, typically for particular clinical populations.

In this context, training a deep network on a large labeled source domain seems to be a good compromise. Yet, applying this method directly to records from a target domain is likely to give lower performances than expected. Indeed cross-dataset

This work was supported by the French Association Nationale de la Recherche et de la Technologie under Grant 2015/1005

generalization is a difficult problem and is referred to as the *dataset bias problem* [18]. Domain adaptation is therefore a crucial step to improve the generalization performance of the model on a target domain.

Our contribution is a benchmark of OTDA approaches versus standard DA methods to enhance generalization performances of "source domain trained" deep sleep stage classifier on a target domain. In this paper, (i) we recall the OTDA principles, (ii) we present the general approach to adapt sleep stage classifier with OTDA, (iii) we perform a benchmark of DA methods on the sleep stage classification task between a large source domain and a target domain. All data are publicly available.

II. Method

Notation: With $n \in \mathbb{N}$ an integer, $\mathbb{1}_n$ (resp. 0_n) stands for the vector of $\mathbb{R}^n$ composed of 1 (resp. 0). Let A, B be two matrices of same dimensions, $\langle A, B\rangle = \mathrm{tr}(A^T B)$. Let $C \in \mathbb{N}$ be the number of possible classes to predict and $\mathcal{Y} = \{1, \cdots, C\}$. $\mathcal{X} = \mathbb{R}^d$ stands for the feature space in dimension d. $\mathcal{X}$ is equipped with the Euclidean norm $\|\cdot\|_2$. For $x \in \mathcal{X}$, δ_x stands for the Dirac measure at position x. Let $n_s, n_t \in \mathbb{N}$ be the number of source training samples and target samples. (X_s, Y_s) (resp. (X_t, Y_t)) stand for the n_s source (resp. n_t target) samples. They are assumed to be uniformly drawn from a source (resp. target) probability distribution μ_s (resp. μ_t). Let $P \in \mathbb{R}_+^{n_s \times n_t}$ be a matrix such that $P\mathbb{1}_{n_t} = \mathbb{1}_{n_s}$ and $P^T\mathbb{1}_{n_s} = \mathbb{1}_{n_t}$. The entropy of $P \in \mathbb{R}^{n_s \times n_t}$ is defined as: $H(P) = -\sum_{i,j=1}^{n_s,n_t} P_{ij}(\log P_{ij} - 1)$. I_y stands for the set of source samples with label y: $I_y = \left\{1 \le i \le n_s : Y_s^i = y\right\}$. Finally, $\|\cdot\|_1^p$ stands for the ℓ_1 norm of a vector to the power of $p \in \mathbb{R}_+$.

A. Domain adaptation with optimal transport

We first recall the principles of OTDA methods as introduced in [10], [11], [13]. It consists in (i) solving a linear program (LP) to obtain a coupling matrix linking the source and target samples, (ii) expressing the transported source samples as a linear combination of the target samples, where the linear combination is given by the coupling matrix.

Consider n_s source samples and n_t target samples, with $X_s, X_t \in \mathcal{X}$ drawn i.i.d. from two probability distributions μ_s and μ_t. Both X_s and X_t are associated with two uniform empirical distributions of probability $\hat{\mu}_s = \dfrac{1}{n_s}\sum_{i=1}^{n_s}\delta_{X_s^i}$ and $\hat{\mu}_t = \dfrac{1}{n_t}\sum_{i=1}^{n_t}\delta_{X_t^i}$. We identify them to the vectors $1/n_s\mathbb{1}_{n_s}$ and $1/n_t\mathbb{1}_{n_t}$. OTDA methods involve a pairwise distance matrix $M \in \mathbb{R}^{n_s \times n_t}$ also called cost matrix or ground metric. It is common to define it as $M_{ij} = \|X_s^i - X_t^j\|_2^2$. Furthermore OTDA methods involve a polytope of possible couplings between the empirical distributions $\hat{\mu}_s$ and $\hat{\mu}_t$: $\Pi(\hat{\mu}_s, \hat{\mu}_t) \stackrel{\text{def}}{=} \left\{P \in \mathbb{R}_+^{n_s \times n_t} : P\mathbb{1}_{n_t} = \hat{\mu}_s, P^T\mathbb{1}_{n_s} = \hat{\mu}_t\right\}$.

The first step consists in finding an optimal coupling P^* which minimizes the transported mass $P \mapsto \langle P, M\rangle$ [10], [11]

$$P^* \in \underset{P \in \Pi(\hat{\mu}_s, \hat{\mu}_t)}{\arg\min} \langle P, M\rangle \ . \tag{1}$$

The second step consists in computing a *transportation mapping* T that transports each source sample into the convex hull of the target samples [10], [11]:

$$T(X_s^i) = \frac{\sum_{j=1}^{n_t} P_{ij}^* X_t^j}{\sum_{j=1}^{n_t} P_{ij}^*} = n_s P_i^* X_t \tag{2}$$

where P_i^* stands for the i-th row of P^*.

B. Entropic regularization

In this paragraph, we recall how the LP part can be solved efficiently using an entropic regularization term. Solving the LP (1) involved in OTDA methods is computationally expensive. To cope with this problem [13] introduced an entropic regularization term to the initial problem. The new optimization problem reads:

$$P^* = \underset{P \in \Pi(\hat{\mu}_s, \hat{\mu}_t)}{\arg\min} \langle P, M\rangle - \epsilon H(P), \epsilon > 0 \ . \tag{3}$$

The benefits of such a regularization are threefold: the problem is now strictly convex with a unique solution, the entropic regularization stabilizes the solution and smooths the transportation plan P^*, and finally the solution of (3) can be obtained by iterative diagonal scaling of the Gibbs kernel $K = \exp(-M/\epsilon)$ associated to M. This step is often referred to as *Sinkhorn-Knopp*'s algorithm [13].

C. Regularization on source labels

The authors of [10], [11] introduced an additional regularization to the OT problem (3) to structure the coupling based on the labels of source samples. It is a regularization defined as $\Omega_c(P) = \sum_{j=1}^{n_t}\sum_{y \in \mathcal{Y}} \|P(I_y, j)\|_1^p$ with $p < 1$ (0.5 in practice). It enforces sparsity in the coupling P so that source samples of the same label are transported onto the same target samples, and additionally source samples of opposite labels are transported onto different target samples. With $\epsilon > 0$ and $\eta > 0$, the optimization problem now reads:

$$P^* = \underset{P \in \Pi(\hat{\mu}_s, \hat{\mu}_t)}{\arg\min} \langle P, M\rangle - \epsilon H(P) + \eta\Omega_c(P) \tag{4}$$

We will refer in this manuscript to this approach as LpL1. As the regularization is non-convex, it is solved iteratively by majorization-minimization [10], [11]

D. General method

To demonstrate the benefits of the OTDA method for a deep learning model we propose the following approach: (i) select samples from the source domains and a record from the target domain (ii) extract the features from the penultimate layer (before the classification layer) for the source data and target data (iii) apply the OTDA method on extracted features to adapt the source features. (iv) retrain the network classification layer on the adapted features (v) predict on the target features.

III. Experiments

a) Data: We used MESA [19] as a source domain which contains EEG records with 3 EEG derivations (Fz-Cz, Cz-Oz, C4) et 2 EOG derivations (EOG left - Fpz, EOG right - Fpz). The target domain was MASS-session 3 [20]. We used 61 records and consider the derivations similar to the source domain derivations: Fz-Cz, Cz-Oz, C4 and EOG left - Fp1, EOG right - Fp1 (since Fpz is not available).

b) Features extraction: We followed the methodology described in [16] for preprocessing. Here, 1200 (resp. 200) randomly chosen source records are used for training (resp. validation i.e. monitoring the progress of training). The deep learning model was the model introduced in [16] with 5 EEG like channels. It was trained by minimizing a cross-entropy loss weighted by the inverse of the stage proportions in order to mimic the balanced sampling used in [16]. The source (resp. target) EEG/EOG 30 s signals were propagated through the network and the activations of the penultimate layer were considered as source (resp. target) features. This leads to a feature space of dimension $d = 600$.

c) Baselines: We compared our approach to various strategies. S stands for not using any adaptation method. CORAL (Correlation Alignment) [5] consists of whitening the source data and coloring them with the target covariance matrix. SA (Subspace Alignment) [6] consists of projecting the source and target data onto two subspaces of dimension $d_{SA} < d$, and aligning the source subspace axes onto the target subspace axes. The hyperparameter d_{SA} was optimized in $\{25, 50, 100\}$ via a grid search. OT (resp. LpL1) has 1 hyperparameter ϵ (resp 2 hyperparameters ϵ, η) that was also optimized by grid search over $\left\{10^{-1}, \cdots 10^{3}\right\}$ (resp. $\left\{10^{-1}, \cdots 10^{3}\right\} \times \left\{10^{0}, \cdots 10^{3}\right\}$).

d) Classifier: We used the classification layer of [16]'s model as a classifier, a fully connected layer with a softmax activation. The input dimension is equal to the feature representation dimension (either d or d_{SA} for SA) and the output dimension is equal to the number of classes $C = 5$. For each DA method, the network is trained on adapted source samples by minimizing the categorical cross entropy with stochastic gradient descent and balanced sampling [16]. The training progress is monitored on the validation source data. An early stopping procedure is used to stop the training when no improvement is observed on the validation set after 5 consecutive pass on the data. Once trained, the model predicts the sleep stages on the (adapted) features of a target record.

e) Cross-validation: The classifier is trained on 2500 (adapted) source training samples (500 per class). 2500 additional (adapted) source validation samples are used to monitor training progress and perform early stopping. The classifier then predicts on an unseen target record. The DA methods are fitted on the source training samples and the considered target record. The process is repeated over the 61 target records.

The hyperparameters of SA, OT and LpL1 are optimized using target validation data. This set is composed of 1000 (adapted) target samples from 10 records. For each set of hyperparameters, (1) the DA method is applied to adapt source training / validation samples, (2) the classifier is trained, (3) it is evaluated on (adapted) target validation data. The set of hyperparameters obtaining the highest performance on validation target data is selected.

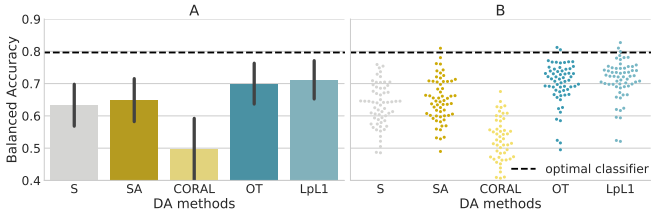

Fig. 1. General benchmark of DA methods: OTDA methods address well the DA problem and recover half of the lost performance. A: averaged balanced accuracy. B: balanced accuracy per record (1 dot is 1 record). Dashed-line: optimal performance of the classifier, should we have labels for 40 records of the target domain.

f) Metrics: The balanced accuracy is used as a general metric. Per-class metrics such as normalized (per row) confusion matrix, F1 score, precision, sensitivity and specificity are furthermore used to characterize the effects of DA methods. Each metric is computed per target record. The presented performances are averaged over the target records.

g) Code: We used POT library (https://pot.readthedocs.io/en/latest) for OT algorithms, PyTorch library [21] for the deep learning part and scikit-learn [22] for general purposes.

h) Benchmark: Besides results with the DA methods, we present the optimal performance that a classifier could reach, should we had access to the labels for 40 records in the target domain. The averaged and per-record balanced accuracies are reported in Figure 1. We observe that OTDA methods outperform SA and CORAL by delivering a gain of performance larger than 5% on the averaged balanced accuracy. Performance per record indicates that most records exhibit higher performances with OTDA methods.

The confusion matrices and the per-class metrics associated to each DA method are reported in Figure 2. They indicate that OTDA methods improve the detection of all stages: the matrices for OT and LpL1 are more diagonal. They also show that OTDA methods offer a good compromise of sensitivity, specificity and precision for each stage. On the other hand, the other DA methods do not exhibit general good performance as they under perform on some particular stages. Indeed SA exhibits low precision and sensitivity on N1 and low precision on REM. CORAL exhibits low sensitivity on N1 and N2 and low precision on N3.

i) Influence of hyperparameters: We focus here on evaluating the influence of the hyperparameters ϵ, η for LpL1. We report the performances in Figure 3-A when varying ϵ in $\left\{10^{-1}, \cdots, 10^{3}\right\}$ and $\eta \in \left\{0, 10^{-1}, \cdots, 10^{3}\right\}$. We also report the performances obtained by setting ϵ, η to the maximum value taken by any feature in either source or target domains. We refer to this as the *maximum heuristic*.

Both ϵ and η have influence on the performances of LpL1 but for each η value, there is a large range of ϵ leading to similar and good performances, see Fig 3-A. The *maximum*

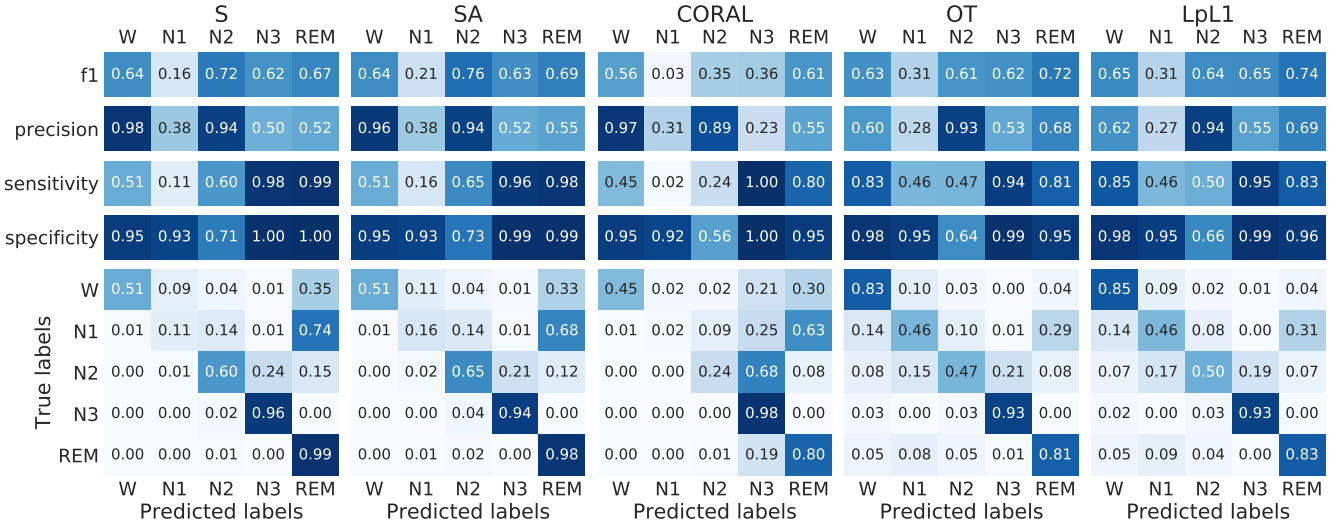

Fig. 2. Per-class metrics of DA methods: OTDA methods improve the detection of all sleep stages regarding the per-class metrics. The darker the better.

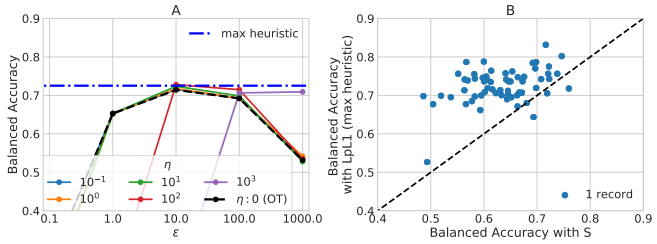

Fig. 3. A: Influence of hyperparameters ϵ, η on LpL1. Dashed-blue-line: ϵ, η fixed with the *maximum heuristic*. For $\eta = 0$, LpL1 = OT. B: Performance improvement per record (each dot is 1 record) with the *maximum heuristic*

heuristic for ϵ, η leads to high performances: the blue dashed-line is always above the η-curves. This also leads to classification improvements on most of the records, see Fig 3-B. Importantly, the *maximum heuristic* avoids the need to run a grid search, and does not require labeled target samples at all.

Conclusion

In this paper, we focused on improving the performances of an automatic sleep stage classifier when applied to a different target domain. The proposed approach based on a convolutional network, and built on optimal transport, clearly showed improvements on cross-domain generalization, a problem which is relevant for any predictive model applied to new datasets exhibiting distribution shifts.

References

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. on Knowl. and Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[2] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *NIPS*, 2006, pp. 137–144.

[3] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Adversarial discriminative domain adaptation," in *CVPR*, 2017.

[4] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette *et al.*, "Domain-adversarial training of neural networks," *JMLR*, vol. 17, no. 59, pp. 1–35, Jan. 2016.

[5] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *AAAI*, 2016, pp. 2058–2065.

[6] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *ICCV*, 2013, pp. 2960–2967.

[7] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *CVPR*, 2012, pp. 2066–2073.

[8] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *ICCV*, 2011, pp. 999–1006.

[9] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," in *IJCAI*, 2009, pp. 1187–1192.

[10] N. Courty, R. Flamary, and D. Tuia, "Domain adaptation with regularized optimal transport," in *ECML/PKDD*, 2014.

[11] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE TPAMI*, 2016.

[12] N. T. H. Gayraud, A. Rakotomamonjy, and M. Clerc, "Optimal Transport Applied to Transfer Learning For P300 Detection," in *BCI Conference*, 2017, pp. 1–6.

[13] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *NIPS*, 2013, pp. 2292–2300.

[14] J. B. Stephansen, A. Ambati, E. B. Leary, H. E. Moore, O. Carrillo, L. Lin *et al.*, "The use of neural networks in the analysis of sleep stages and the diagnosis of narcolepsy," *arXiv:1710.02094*, 2017.

[15] C. Iber, S. Ancoli-Israel, A. Chesson, and S. F. Quan, *The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications*. American Academy of Sleep Medicine, 2007.

[16] S. Chambon, M. N. Galtier, P. J. Arnal, G. Wainrib, and A. Gramfort, "A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series." *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2018.

[17] S. Biswal, J. Kulas, H. Sun, B. Goparaju, M. B. Westover, M. T. Bianchi *et al.*, "SLEEPNET: automated sleep staging system via deep learning," *arXiv:1707.08262*, 2017.

[18] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR*, 2011, pp. 1521–1528.

[19] D. A. Dean, A. L. Goldberger, R. Mueller, M. Kim, M. Rueschman, M. D. *et al.*, "Scaling up scientific discovery in sleep medicine: The national sleep research resource," *Sleep*, vol. 5, p. 11511164, 2016.

[20] C. O'Reilly, N. Gosselin, J. Carrier, and T. Nielsen, "Montreal Archive of Sleep Studies: an open-access resource for instrument benchmarking and exploratory research," *Journal of Sleep Research*, vol. 23, no. 6, pp. 628–635, 2014.

[21] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito *et al.*, "Automatic differentiation in PyTorch," in *NIPS Workshop*, 2017.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel *et al.*, "Scikit-learn: Machine learning in Python," *JMLR*, vol. 12, pp. 2825–2830, 2011.