We are not All Equal: Personalizing Models for Facial Expression Analysis with Transductive Parameter Transfer

Enver Sangineto*, Gloria Zen*
University of Trento
Trento, Italy

Elisa Ricci
FBK Trento, Italy
University of Perugia
Perugia, Italy

Nicu Sebe
University of Trento
Trento, Italy

ABSTRACT

Previous works on facial expression analysis have shown that person-specific models are advantageous with respect to generic ones for recognizing facial expressions of new users added to the gallery set. This finding is not surprising, due to the often significant inter-individual variability: different persons have different morphological aspects and express their emotions in different ways. However, acquiring person-specific labeled data for learning models is a very time consuming process. In this work we propose a new transfer learning method to compute personalized models without labeled target data. Our approach is based on learning multiple person-specific classifiers for a set of *source* subjects and then directly transfer knowledge about the parameters of these classifiers to the *target* individual. The transfer process is obtained by learning a regression function which maps the data distribution associated to each source subject to the corresponding classifier's parameters. We tested our approach on two different application domains, Action Units (AUs) detection and spontaneous pain recognition, using publicly available datasets and showing its advantages with respect to the state-of-the-art both in term of accuracy and computational cost.

Categories and Subject Descriptors

I.2.10 [**Vision and Scene Understanding**]: [Video Analysis]; H.1.2 [**User/Machine Systems**]: [Human factors, Human information processing]

General Terms

Human factors

Keywords

Facial Expression Recognition, Action Unit Detection, Transductive Transfer Learning, Learning from Distributions

*These two authors contributed equally to this work

MM'14, November 3–7, 2014, Orlando, Florida, USA.

http://dx.doi.org/10.1145/2647868.2654916 .

1. INTRODUCTION

Social interaction among individuals is regulated by a complex communication code. Everyday people naturally sense and convey social signals (*e.g.* interest, approval, disagreement). Automatic understanding this communication code is one of the priorities for the interpretation of human social activities. Aiming to face the challenges of creating socially aware computers, in the last few years the research in the area of social signal processing [34] has made considerable progresses. Social signals are conveyed (also) through a great variety of non-verbal behavioral cues such as facial expressions, gaze, gestures, body postures and vocal outbursts. Automatic detection of non-verbal behavioral cues and specifically of facial expressions [19, 37] is of crucial importance in many areas, including ambient assisted living, human-computer interfaces, entertainment, education, and multimedia content analysis. For instance, in [36] facial expressions are employed to estimate the level of interest of a user watching a video, in [15] they are used for group meeting analysis and in [30] for affective labeling of multimedia contents. However, human beings express their emotions in different ways. Cultural factors, age, gender and personality strongly influence the intensity and the way in which emotions are exhibited [37]. Other important variability factors are the person-specific morphological appearance as well as the illumination conditions or the camera viewpoint.

State-of-the-art facial expression analysis systems achieve excellent performance in controlled laboratory conditions, with limited illumination variations, frontal head pose and posed expressions. However, the accuracy of these systems often drastically drops in real world situations with spontaneous expressions. This drop is largely due to the fact that the datasets used for training models do not sufficiently well represent the variability of real-world scenarios. Technically, this issue can be seen as a special case of the well known *dataset bias* effect [14]: the assumption that training and test data are drawn from the same distribution often does not hold in realistic conditions and the performance of a classifier typically degrades when tested in a different setting from the training one. On the other hand, learning person-specific or scenario-specific classifiers is often practically infeasible, due to the need of collecting large number of task-specific labeled samples. This problem has recently motivated a large series of efforts in the computer vision and multimedia communities in developing transfer learning and domain adaptation approaches [25, 35]. These methods are specifically designed to exploit the informations acquired in

different but related *source* domains/tasks when training a classifier for a novel *target* task or domain.

However, few attempts have been done so far for using these techniques in the context of facial expression analysis and in particular to learn *person-specific* models [4, 5]. In [4] Chen *et al.* proposed a transfer learning approach for spontaneous pain recognition. Their method can be used both in an *inductive* and in a *transductive* setting. In both cases labeled data are provided for a set of individuals (source). In the inductive setting, few labeled samples are also available for the subject of interest, while in the more challenging transductive scenario no annotated data for the target individual are collected. In [5] a transductive transfer learning approach named Selective Transfer Machine is introduced for AU detection. These methods provide significant advantages over generic classifiers in term of recognition accuracy. However, their computational cost is usually not negligible when large datasets are considered as they rely on re-weighting all source samples according to their similarity with target data.

In this paper we present a radically different knowledge transfer strategy to learn personalized models for facial expression analysis in a transductive setting (Fig. 1). Transfer learning is formulated as the problem of learning from a set of data distributions. Specifically, we assume to have at disposal N source datasets. Every source domain corresponds to a specific individual. Source data are labeled (*e.g.* indicating an emotion type or a specific AU). Our goal is to learn a *person-specific* classifier for a new subject (target) improving the accuracy over the *generic* classifier, *i.e.* the classifier trained only on source samples. Data in the target domain are not annotated. In our approach, first a set of N classifiers, parameterized by the vectors $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N$, is obtained from source data. In a second phase, we propose to use a vector-valued regression framework to learn a mapping $f(\cdot)$ between each source data distribution and the parameters of the associated classifier $\boldsymbol{\theta}_i$. Once $f(\cdot)$ is obtained, it is used to predict the optimal classifier $\boldsymbol{\theta}_t$ for the target distribution.

Our method has been applied to two different problems, facial AUs detection and pain expression recognition, and evaluated using two publicly available datasets: the Extended Cohn-Kanade dataset [17] and the UNBC-McMaster Shoulder Pain Expression Archive Database [18]. Our experiments demonstrate the advantages of the proposed approach over state-of-the-art generic classifiers, confirming the findings of previous works proposing personalized models. When comparing with methods based on transfer learning [4, 5], our approach achieves similar or better recognition accuracy but radically outperforms them in term of computational cost.

Contributions. To summarize, the main contributions of this paper are the following: (i) We propose a novel transfer learning approach to obtain personalized models for facial expression analysis. To the best of our knowledge, this is the first approach for *Transductive Parameter Transfer (TPT)*, where the parameters of the source classifiers are "transferred" to the target domain using a regression framework without the need of labeled target data. Previous methods either rely on *instance transfer* (source sample selection or re-weighting) or look for a *shared feature space* between sources and target data [21]. (ii) Our approach is computationally very efficient both at training and at testing time, a crucial aspect in many applications (*e.g.* HCI). (iii) In this work, we assume that the different domains correspond to different users expressing the same emotions but our approach can be applied to cope with many other variability sources such as, for instance, viewpoint changes.

2. RELATED WORK

Facial Expression Analysis. In the last few years research on facial expression analysis have made significant progresses. Many approaches have proved to be effective for recognizing the seven basic facial expressions (*i.e.* happiness, sadness, fear, anger, disgust, and surprise plus the neutral state) or for detecting AUs on which the facial emotions are based [19, 37]. The common approach adopted in the state-of-the-art face analysis systems for still images consists of three main steps: face registration, feature extraction (possibly followed by dimensionality reduction) and classification. Face registration is commonly achieved by the localization of anatomically salient facial points [27, 28]. Once the face has been registered, different features can be extracted either from the whole face image (*e.g.* Local Binary Pattern Histograms [1]) or from patches centered around (a subset of) the facial landmarks (*e.g.* SIFT features). Concerning the classification step, SVM, Boosting, Random Forests and many other approaches have been proposed [19, 37]. However, most state-of-the-art systems have been trained and tested in laboratory conditions, with datasets mainly composed of frontal face images and posed emotions [19, 32, 37]. Much little attention has been payed to personalized systems and realistic scenarios. Many authors have recently focused on recognizing spontaneous facial emotions [7, 16, 33], while others have focused on the recognition of non-basic emotions such as pain or frustration [13, 29]. However, all these works are based on generic detectors, *i.e.* detectors trained using a dataset as much as possible realistic and which is supposed to generalize to different individuals and acquisition conditions. Unfortunately, having at disposal only datasets of few hundreds/thousands of images, generalization is hard to achieve.

To cope with this issue few recent works have proposed solutions to integrate weakly labeled or unlabeled data. In [29] Sikka *et al.* adopted a Multiple Instance Learning approach for training a pain expression classifier using video-level labels assuming that frame-level labels are not available. Pain/no-pain expression classification in a partially labeled and unlabeled setting is also considered in [4] where an extension of AdaBoost is proposed. However, their method did not achieve significant improvement in terms of accuracy with respect to the generic classifier in the transductive scenario. In [5] Chu *et al.* presented Selective Transfer Machine for person-specific AU detection. Their approach is based on the Kernel Mean Matching technique [9], which is modified using an iterative minimization procedure where labeled source data drive a progressive movement of the generic SVM hyperplane toward the target space. Even if effective, this approach is very slow at training time, being the proposed minimization strategy very time consuming. On the other hand, user-specific adaptation algorithms are required to be computationally efficient to be used in real world applications. Our method is mainly motivated based on this need and our experiments confirm that it is particularly advantageous with respect to [5].

Transfer learning. Transfer learning approaches have recently become popular as a means to solve or alleviate the scarceness of (labeled) training samples [35, 25]. Since a complete survey of transfer learning techniques is beyond the scope of this article, we refer the reader to [21] where a taxonomy of the existing approaches is presented showing how most of them can be classified in instance-transfer, features-transfer and parameter-transfer approaches.

When the target domain is completely unlabeled, usually instance-transfer approaches are commonly adopted. For instance, in *covariate shift* problems, the basic assumption is that the source and the target feature spaces are the same but the marginal probability distributions of the input data are different [21]. In [9] Gretton *et al.* compare the centroids of the source and the target distributions and estimate the source sample weights which reduce this discrepancy. These weights are then used to assign importance to the source samples when training a model on target data.

There are two main drawbacks with instance-transfer approaches. The first is the computational burden. All the source samples need to be stored as they will be selected or re-weighted when training a model on target data. This is not very efficient both in terms of memory requirements and computational cost. The second issue is that computing the distance between the means of the source and the target data distributions in the Reproducing Kernel Hilbert Space as in many previous works [5, 9] may poorly approximate the real difference between distributions. We propose to solve both problems by learning the relation between the "shape" of each task-specific data distribution and the corresponding classifiers' parameters. Once this mapping has been learned using source data, source samples can be completely discarded. In Sec. 3 we show that we only need to store a kernel matrix representing the distribution similarity. Moreover, we show that our method is flexible as it permits to define several measures to compare source and target data distributions, each corresponding to a specific kernel function.

Learning from Distributions. There is a long track of works on learning from data distributions, showing successful results in several fields, such as text analysis [11], bioinformatics [2], and computer vision [8]. The common approach is to map a distribution into a Reproducing Kernel Hilbert Space, introduce a suitable kernel function, and then use a traditional kernel machine. In this context, the large majority of methods [10, 11] operates by fitting a parametric density function (*e.g.* a Gaussian distribution) to the training data, and using the density parameters to compute the inner products between distributions. Another class of methods does not make any assumption on the distribution form and defines kernels among sets of objects [8]. To our knowledge, no previous works have considered kernel among distributions to perform domain adaptation. An interesting exception is [2], where Blanchard *et al.* proposed a learning framework based on kernel machines and the kernel matrix is the product between two terms: a "standard" kernel defined on feature vectors is multiplied by a kernel defined on the corresponding pairs of distributions. However in [2] there is no regression framework or explicit transfer of the source classifier's parameters to the target domain. Finally, the solution proposed by Blanchard and colleagues involves storing and comparing all the source and target samples which gives no computational advantages with respect to other techniques like instance-reweighting.

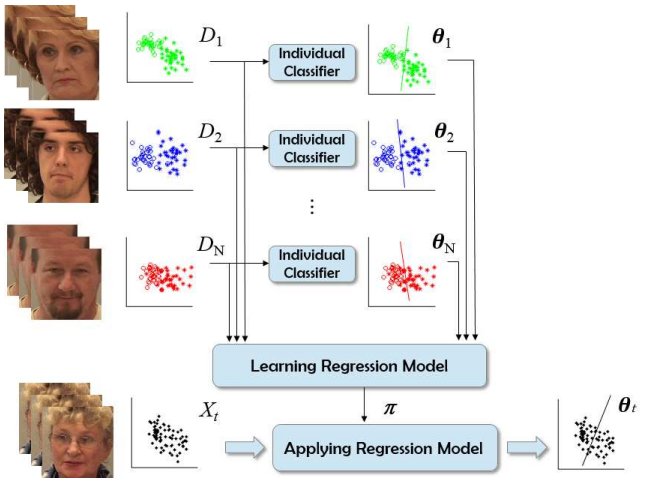

Figure 1: Overview of the proposed *Transductive Parameter Transfer (TPT)* learning approach for facial expression analysis.

3. A REGRESSION FRAMEWORK FOR PARAMETER TRANSFER

In this section we present our method for Transductive Parameter Transfer. Let $\mathcal{X}, \mathcal{Y}$ be, respectively, a feature space and a label space. In this paper we consider $\mathcal{Y} = \{-1, 1\}$ but generalizing our approach to a multiclass setting is straightforward. We assume that N *labeled source* datasets $D_1^s, ..., D_N^s$, $D_i^s = \{\mathbf{x}_j^s, y_j^s\}_{j=1}^{n_i^s}$, and an *unlabeled target* dataset $X^t = \{\mathbf{x}_j^t\}_{j=1}^{n_t}$, $\mathbf{x}_j^s, \mathbf{x}_j^t \in \mathcal{X}$, $y_j^s \in \mathcal{Y}$, are available. Moreover, let denote $X_i^s = \{\mathbf{x}_j^s\}_{j=1}^{n_i^s}$ the set of points in D_i^s obtained by discarding the labels. We assume that the elements in X_i^s are generated by a marginal distribution P_i^s defined on $\mathcal{X}$ and similarly the vectors X^t are generated by P_t. We generally assume that $P_t \neq P_i^s$ and $P_i^s \neq P_j^s$ $(1 \leq i, j \leq N,\ i \neq j)$. Finally, we call $\mathcal{P}$ the space of all the possible distributions on $\mathcal{X}$ and we assume that P_t, P_i^s are drawn from $\mathcal{P}$ according to the meta-distribution Π, *i.e.* $P_t, P_i^s \sim \Pi$.

Our goal is to learn a classifier on the target data X^t without acquiring label information. The approach we propose is based on three main steps (Fig. 1). First, a set of source-specific classifiers is learned on each training set D_i^s. In the second step a regression algorithm is adopted in order to learn the relation between the marginal distributions P_i^s and the source classifiers' parameter vectors $\boldsymbol{\theta}_i$. Finally, the desired target classifier is obtained applying the learned distribution-to-classifier mapping and using as input the distribution P_t. In the following, the three steps are described in details.

In the first phase, each source dataset D_i^s is used to train a classifier solving an optimization problem as:

$$\boldsymbol{\theta}_i = \min_{\boldsymbol{\theta} \in \Theta} \mathcal{R}(\boldsymbol{\theta}) + \lambda_L L(\boldsymbol{\theta}, D_i^s) \tag{1}$$

where $\boldsymbol{\theta}_i$ is the parameter vector associated to the learned classifier, Θ is the parameter space, $\mathcal{R}(\cdot)$ a regularizer and $L(\cdot)$ is the empirical risk weighted with λ_L. Specifically,

in this paper we use a set of linear SVM classifiers, thus $\boldsymbol{\theta}_i = [\mathbf{w}_i',\ b_i]$, $\mathbf{w}_i \in I\!R^M, b_i \in I\!R$, defines a hyperplane in the feature space $\mathcal{X} \equiv I\!R^M$ (see Fig. 1) and it can be estimated solving:

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 + \lambda_L \sum_{j=1}^{n_i^s} l(\mathbf{w}'\mathbf{x}_j^s + b, y_j^s) \tag{2}$$

where $l(\cdot)$ is the hinge loss.

In the second phase, we propose to use a regression approach to learn a mapping $f : \mathcal{P} \to \Theta$. The intuition here is that each hyperplane, defined by $\boldsymbol{\theta}_i$, depends on the distribution P_i^s generating the datapoints X_i^s. Thus, if using the source data we are able to learn the relationship between the "shape" of the underlying distribution and its corresponding hyperplane, then, for computing the optimal hyperplane on target data, we do not need label information anymore and we can simply apply the learned mapping $f(\cdot)$, *i.e.* $f(P_t) = [\mathbf{w}_t',\ b_t]$.

Of course, we do not know P_i^s $(1 \leq i \leq N)$ neither P_t, but we assume they can be approximated with the associated sample sets X_i^s and X^t, respectively. Thus, given a training set $\mathcal{T} = \{X_i^s, \boldsymbol{\theta}_i\}_{i=1}^N$ we propose to learn a mapping:

$$\widehat{f} : 2^{\mathcal{X}} \to \Theta \tag{3}$$

which approximates $f(\cdot)$. The function $\widehat{f}(\cdot)$ is a vector-valued set function, *i.e.* a function which takes as input a set of datapoints X and outputs a vector $\boldsymbol{\theta} = [\mathbf{w}',\ b]$. If $\mathcal{X} \equiv I\!R^M$, *i.e.* $\boldsymbol{\theta}$ is a $M+1$ dimensional vector, $\widehat{f}(\cdot)$ can be learned using $M+1$ independent scalar regressors. However, as it is reasonable to assume that the elements in $\boldsymbol{\theta}$ are correlated, we use a vector-valued regression approach where the output dimensions are jointly estimated. In our implementation, we adopt the Multioutput Support Vector Regression (M-SVR) framework proposed in [31]. In preliminary experiments we also tested a k-nearest neighbour approach, averaging the $\boldsymbol{\theta}$ values corresponding to the k-closest sources, based on the kernels presented in the next sections, but we obtained much worse results with respect to the proposed Multioutput SVR. We believe that this is due to the fact that SVR is a better tool for regression.

The M-SVR is a generalization of the ϵ-insensitive Support Vector Regression to a multi-dimensional case. In the M-SVR framework, $\widehat{f}(\cdot)$ can be defined by a set of parameters $\boldsymbol{\pi} = (\mathbf{B}, \mathbf{c})$:

$$\widehat{f}(X) = \phi(X)'\mathbf{B} + \mathbf{c}' \tag{4}$$

where $\mathbf{B} = [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{M+1}]$ and $\mathbf{c} = [c_1, \ldots, c_{M+1}]'$ are the weight matrix and the bias vector, respectively, and $\phi(X)$ is a nonlinear mapping of the set of data X to a higher-dimensional space. In turn $\boldsymbol{\pi}$ can be found by minimizing:

$$\min_{\boldsymbol{\pi}} \frac{1}{2} \sum_{i=1}^{M+1} ||\boldsymbol{\beta}_i||^2 + \lambda_E \sum_{i=1}^{N} E(\|\boldsymbol{\theta}_i' - \widehat{f}_{\boldsymbol{\pi}}(X_i)\|) \tag{5}$$

where $E(\cdot)$ is a loss function which extends to the multi-dimensional case the ϵ-insensitive loss proposed by Vapnik for scalar-valued Support Vector Regression, *i.e.*:

$$E(u) = \begin{cases} 0 & u < \epsilon \\ u^2 - 2u\epsilon + \epsilon^2 & u \geq \epsilon \end{cases} \tag{6}$$

As for scalar-valued SVR, the problem (5) can be solved in its dual form introducing the kernel matrix $\mathbf{K} \in I\!R^{N \times N}$,

Algorithm 1 Optimization algorithm to solve (5)

Input: The set $\mathcal{T} = \{X_i^s, \boldsymbol{\theta}_i\}_{i=1}^N$, the parameters λ_E, ϵ.

Initialize $k = 0$, $\mathbf{V}^k = \mathbf{0}$, $\mathbf{c}^k = \mathbf{0}$.
Inner Loop:
Compute a_i using (9), $i = 1, \ldots, N$.
Compute $\hat{\mathbf{V}}$, $\hat{\mathbf{c}}$ solving (8) $\forall\ j = 1, \ldots, M+1$.
Compute η_k using a backtracking algorithm.
Compute $\mathbf{V}^{k+1} = \mathbf{V}^k + \eta_k(\hat{\mathbf{V}} - \mathbf{V}^k)$.
Compute $\mathbf{c}^{k+1} = \mathbf{c}^k + \eta_k(\hat{\mathbf{c}} - \mathbf{c}^k)$.
Set $k = k + 1$.
Until Convergence

Output: $\mathbf{V}, \mathbf{c}$

Algorithm 2 The proposed TPT approach

Input: The sets $D_1^s, ..., D_N^s$, X^t, the regularization parameters λ_L, λ_E, ϵ.

Compute $\{\boldsymbol{\theta}_i = (\mathbf{w}_i, b_i)\}_{i=1}^N$ using (2).
Create a training set $\mathcal{T} = \{X_i^s, \boldsymbol{\theta}_i\}_{i=1}^N$.
Compute the kernel matrix $\mathbf{K}$, $\mathbf{K}_{ij} = \kappa(X_i^s, X_j^s)$ using (10), (13) or (14)
Given $\mathbf{K}$, $\mathcal{T}$, compute $\widehat{f}(\cdot)$ solving (5).
Compute $(\mathbf{w}_t, b_t) = \widehat{f}(X^t)$ using (7).

Output: $\mathbf{w}_t, b_t$

$\mathbf{K}_{ij} = \kappa(X_i, X_j) = \phi(X_i)'\phi(X_j)$ [31] and the decision function (4) can be rewritten as:

$$\widehat{f}(X) = \sum_{i=1}^{N} \mathbf{V}_{i\cdot}\kappa(X_i, X) + \mathbf{c}' \tag{7}$$

where $\mathbf{V} \in I\!R^{N \times M+1}$ is the matrix of the optimal parameters computed solving the dual optimization problem associated to (5) and $\mathbf{V}_{i\cdot}$ denotes the i-th row. To compute $\mathbf{V}$ and $\mathbf{c}$ in this paper we follow [31] and adopt an iterative reweighted least-squares procedure. This procedure is summarized in Algorithm 1. We define the matrix $\boldsymbol{\Theta} \in I\!R^{N \times M+1}$ $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N]'$. At each iteration k, the values of $\mathbf{V}$ and $\mathbf{c}$ are updated solving a series of $M+1$ independent weighted least-squares problems, one for each column of $\mathbf{V}$ (here denoted as $\mathbf{V}_{\cdot j}$) and for each c_j:

$$\begin{bmatrix} \mathbf{K} + \mathbf{A} & \mathbf{1} \\ \mathbf{a}'\mathbf{K} & \mathbf{1}'\mathbf{a} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{\cdot j} \\ c_j \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Theta}_{\cdot j} \\ \mathbf{a}\boldsymbol{\Theta}_{\cdot j} \end{bmatrix} \tag{8}$$

where $\boldsymbol{\Theta}_{j\cdot}$ is the j-th row of the matrix $\boldsymbol{\Theta}$ and $\mathbf{1}$ is an all-one column vector. The vector $\mathbf{a} = [a_1, \ldots, a_N]$ and the matrix $\mathbf{A} \in I\!R^{N \times N}$, $\mathbf{A}_{ij} = a_i\delta(i-j)$ are computed at each step using:

$$a_i = \begin{cases} 0 & u_i^k < \epsilon \\ \frac{2\lambda_E(u_i^k - \epsilon)}{u_i^k} & u_i^k \geq \epsilon \end{cases} \tag{9}$$

and $u_i^k = \|\boldsymbol{\theta}_i' - \sum_{i=1}^N \mathbf{V}_{i\cdot}^k \kappa(X_i, X) - (\mathbf{c}^k)'\|$. Due to lack of space, for more details on the M-SVR framework we refer the reader to the original paper [31].

Finally, once (5) is solved, in the last phase of our method, (7) is used to compute the parameter vector of the target classifier, *i.e.* $\boldsymbol{\theta}_t = \widehat{f}(X^t)$.

From Algorithm 1 and from (7) it is clear that, both for computing $\widehat{f}(\cdot)$ and to apply $\widehat{f}(\cdot)$ to X^t, we only need a kernel $\kappa(X_i, X_j)$ which represents the similarity between pairs of datasets X_i, X_j. In Sec. 3.1-3.3 we propose different choices for such a kernel, but it is worth noting that other kernels can be used as well.

In Algorithm 2 we summarize the whole *training* procedure. The *test* phase is standard as in classification with SVM. Given a new target feature vector $\mathbf{x}$, the corresponding label y is predicted using: $y = sign(\mathbf{w}_t'\mathbf{x} + b_t)$.

3.1 Fisher Kernel

Fisher kernels [10], originally proposed in machine learning and statistics to measure the similarity between distributions, have recently become common tools in the computer vision and multimedia fields [20, 22].

Suppose that the set of points $X = \{\mathbf{x}_t\}_{t=1}^n$ is generated by the marginal distributions P on $\mathcal{X}$. Let p_γ be a probability density function which models the generative process of elements in X where γ is the parameter vector governing p_γ. In statistics, the score function is defined as $G_\gamma = \nabla_\gamma \log p_\gamma(X)$, *i.e.* it is the gradient of the log-likelihood of the data with respect to the model parameters and describes how the parameters of the generative model p_γ should be modified to better fit the data [10]. Typically p_γ is chosen as a Gaussian Mixture Model and $\gamma = \{\alpha_h, \mu_h, \Sigma_h, h = 1, ..., H\}$, being H the number of components and $\alpha_h, \mu_h, \Sigma_h$ the component weight, its mean and its covariance matrix, respectively. In our experiments we set $H = 20$. As it is common, we also assume that every matrix Σ_h is diagonal, *i.e.* $\Sigma_h = \text{diag}(\sigma_h)$.

Given two sets of points X_i and X_j generated by the two distributions P_i and P_j, their similarity can be measured using the Fisher Kernel [10]:

$$\kappa_{FK}(X_i, X_j) = (G_\gamma^{X_i} Z_\gamma)' Z_\gamma G_\gamma^{X_j} = (\mathcal{G}_\gamma^{X_i})' \mathcal{G}_\gamma^{X_j} \tag{10}$$

where $F_\gamma = Z_\gamma' Z_\gamma$ is the Cholesky decomposition of the Fisher Information Matrix [10] and $\mathcal{G}_\gamma^X$ is the so called Fisher vector. The Fisher vector [22] is obtained computing $\mathcal{G}_\gamma^X = [\mathcal{G}_{\alpha_1}^X, \ldots, \mathcal{G}_{\alpha_H}^X, \mathcal{G}_{\mu_1}^X, \ldots, \mathcal{G}_{\mu_H}^X, \mathcal{G}_{\sigma_1}^X, \ldots, \mathcal{G}_{\sigma_H}^X]$, *i.e.* calculating and concatenating the following terms ($\forall h = 1, \ldots, H$):

$$\begin{aligned}
\mathcal{G}_{\alpha_h}^X &= \frac{1}{\sqrt{\omega_h}} \sum_t (\psi_t(h) - \omega_h) \\
\mathcal{G}_{\mu_h}^X &= \frac{1}{\sqrt{\omega_h}} \sum_t \psi_t(h) \frac{\mathbf{x}_t - \mu_h}{\sigma_h} \\
\mathcal{G}_{\sigma_h}^X &= \frac{1}{\sqrt{2\omega_h}} \sum_t \psi_t(h) \left[\frac{(\mathbf{x}_t - \mu_h)^2}{\sigma_h^2} - 1 \right]
\end{aligned} \tag{11}$$

where $\omega_h = \frac{\exp(\alpha_h)}{\sum_j \exp(\alpha_j)}$ and $\psi_t(h)$ represents the soft assignment of $\mathbf{x}_t$ to the h-th Gaussian. We refer to [22] for further details.

3.2 EMD-based kernel

The Earth Mover's Distance [26, 24] has been widely used in computer vision as it represents a simple and practical approach to measure the distance between distributions. To compute the EMD between X_i and X_j, first a clustering algorithm is applied separately to the two datasets (we use a simple k-means algorithm in our experiments). In this way the signatures of each set $\mathcal{I} = \{(\boldsymbol{\nu}_1^i, w_1^i), \ldots (\boldsymbol{\nu}_Q^i, w_Q^i)\}$ and $\mathcal{J} = \{(\boldsymbol{\nu}_1^j, w_1^j), \ldots (\boldsymbol{\nu}_Q^j, w_Q^j)\}$ are computed, where $\boldsymbol{\nu}_q^i$, $\boldsymbol{\nu}_q^j$ are the cluster centroids respectively obtained on the X_i and X_j datasets and w_q^i, w_q^j denote the weights associated to each cluster. In this paper, for sake of simplicity, we consider the same number of clusters $Q = 20$ for both datasets and the cardinality of each cluster is used as cluster weight.

Given two signatures $\mathcal{I}$ and $\mathcal{J}$, the EMD between the associated datasets X_i and X_j is defined as the solution of the following transportation problem:

$$\begin{aligned}
D_{EMD}(X_i, X_j) = &\min_{f_{pq} \geq 0} \sum_{p,q=1}^{Q} d_{pq} f_{pq} \\
\text{s.t.} \quad &\sum_{p=1}^{Q} f_{pq} = w_q^i \quad \sum_{q=1}^{Q} f_{pq} = w_p^j
\end{aligned} \tag{12}$$

where f_{pq} are flow variables and d_{pq} is the ground distance defined as $d_{pq} = \|\boldsymbol{\nu}_p^i - \boldsymbol{\nu}_q^j\|$. In a nutshell, the EMD represents the minimum cost needed to transform one distribution into another. Using EMD we define a kernel:

$$\kappa_{EMD}(X_i, X_j) = e^{-\rho D_{EMD}(X_i, X_j)} \tag{13}$$

where ρ is a user defined parameter. Despite this is not a valid kernel as it is not semi-definite positive we observe excellent performance in our experimental evaluation. This is in line with the findings of previous works [6].

3.3 Density Estimate-based Kernel

The last choice for a kernel measuring the similarity of two distributions we present here is taken from [2]. It is based on a Density Estimate (DE) kernel and it is defined as follows:

$$\kappa_{DE}(X_i, X_j) = \frac{1}{nm} \sum_{p=1}^{n} \sum_{q=1}^{m} \kappa_{\mathcal{X}}(\mathbf{x}_p, \mathbf{x}_q), \tag{14}$$

where n, m are the cardinality of X_i, X_j, respectively, and $\kappa_{\mathcal{X}}(\cdot)$ is a normalized gaussian kernel defined on $\mathcal{X}$.

4. EXPERIMENTS

In this section we present two series of experiments to demonstrate the effectiveness of the proposed TPT approach in two different application domains: AU detection (Sec. 4.1) and pain recognition (Sec. 4.2). We deliberately choose these in order to compare our approach with the only other transductive domain adaptation methods for facial expression analysis we are aware of: the Transductive AdaBoost (TA) algorithm proposed in [4] and the Selective Transfer Machine (STM) in [5]. For this reason, we adopt the same experimental protocols presented in [4, 5], *i.e.* we use the same benchmarks, the same image registration and features extraction pipelines and the same evaluation scheme (on a frame basis) and metrics (Area Under ROC and/or F_1 Score). In the following subsections we present the results of our experimental evaluation in details. Note that in [5] the authors also use the RU-FACS dataset which is no more publicly available. Therefore, we could not test our system on that benchmark. We conclude this section analyzing the

Table 1: Performance on Cohn-Kanade+ dataset, F_1 Score

AU	SVM	KMM [9]	TSVM [12]	DASVM [3]	STM [5]	TPT EMD	TPT Fisher	TPT DE
1	61.1	44.9	56.8	57.7	62.2	72.2	74.0	74.4
2	73.5	50.8	59.8	64.3	76.2	81.8	75.5	84.2
4	62.7	52.3	51.9	57.7	69.1	71.5	71.8	66.3
6	75.7	70.1	47.8	68.2	79.6	75.1	74.9	74.8
12	76.7	74.5	59.6	59.0	77.2	85.5	83.5	85.1
17	76.0	53.2	61.7	81.4	84.3	82.8	83.5	76.1
Avg	70.9	57.6	56.3	64.7	74.8	**78.2**	77.2	76.8

Table 2: Performance on Cohn-Kanade+ dataset, AUC.

AU	SVM	KMM [9]	TSVM [12]	DASVM [3]	STM [5]	TPT EMD	TPT Fisher	TPT DE
1	79.8	68.9	69.9	72.6	88.9	88.0	89.0	88.2
2	90.8	73.5	69.3	71.0	87.5	93.5	92.9	92.6
4	74.8	62.2	63.4	79.9	81.1	88.1	85.0	84.3
6	89.7	87.7	60.5	94.7	94.0	92.2	91.3	91.1
12	88.1	89.5	76.0	95.5	92.8	97.5	97.2	97.1
17	90.3	66.6	73.1	94.7	96.0	95.9	94.3	94.3
Avg	85.6	74.7	68.7	83.1	90.1	**92.5**	91.6	91.3

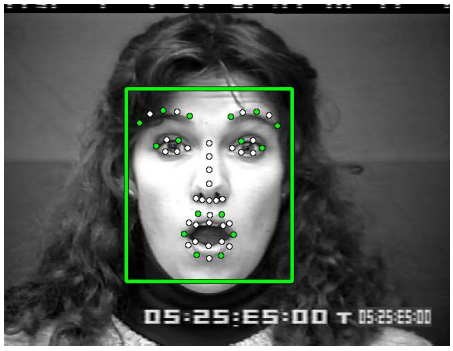

Figure 2: CK+ dataset: sample frame. The green rectangle shows the cropped part of the image and the dots are the detected facial landmarks (the 16 selected landmarks are in green).

influence of the cardinality of the source datasets (N) and the number of target samples (n_t) on the accuracy of the proposed method (Sec. 4.3).

4.1 Facial Action Unit Detection

4.1.1 Dataset

The **Extended Cohn-Kanade**[1] (CK+) dataset [17] contains a set of spontaneous and posed expressions with only frontal faces. The dataset includes 593 videos from 123 users. The number of videos per user ranges from 1 to 11. The video length varies from 4 to 71 frames. A sample frame extracted from this dataset is shown in Fig. 2.

4.1.2 Feature Extraction

In order to compare our results with [5], we followed the same experimental protocol and we implemented the same feature extraction pipeline. First, the subject's face and facial landmarks are detected, the face is aligned, cropped and resized to a 200×200 pixel window. We use the code available at the author's website[2]. Then, as in [5] we select 16 landmarks (Fig. 2) from which we extract SIFT descriptors using OpenCV from 36×36 pixel regions around them. In [5] the authors do not specify what landmarks to select, so we chose the external corners of the mouth, of the eyes and of the eyebrows plus other 10 equally spaced landmarks in the mouth, the eyes and the eyebrows. Finally, all these descriptors are concatenated and dimensionality is reduced using Principal Component Analysis. We retain 90% of the energy, obtaining a final feature vector of size 51. Similarly to [5], we select the most frequent AUs in the dataset and the detection of each AU is considered as an independent binary classification problem. However, differently from [5], we could not test our system on AU7 and AU15 because the number of samples for these two AUs is too small. This fact, in combination with the low number of persons (N) in the CK+ dataset, makes it difficult for our system to learn the relation between the data point distributions and the hyperplane parameters. Thus, the average performance values showed in Tables 1-2 are computed using 6 AUs.

Figure 3: UNBC-MSPEAD dataset: sample frames. Spontaneous facial expressions of patients under shoulder mobility tests are shown.

4.1.3 Results

Following [5], our experiments are conducted using a leave-one-subject-out evaluation scheme. The performance are computed using the F_1 score, defined as $F_1 = \frac{2 \cdot recall \cdot precision}{recall + precision}$, and the Area Under ROC (AUC).

We compare our approach with a generic classifier learned on the entire source data (SVM), a semi-supervised Trans-

[1] http://www.pitt.edu/∼emotion/ck-spread.htm

[2] http://humansensing.cs.cmu.edu/intraface/

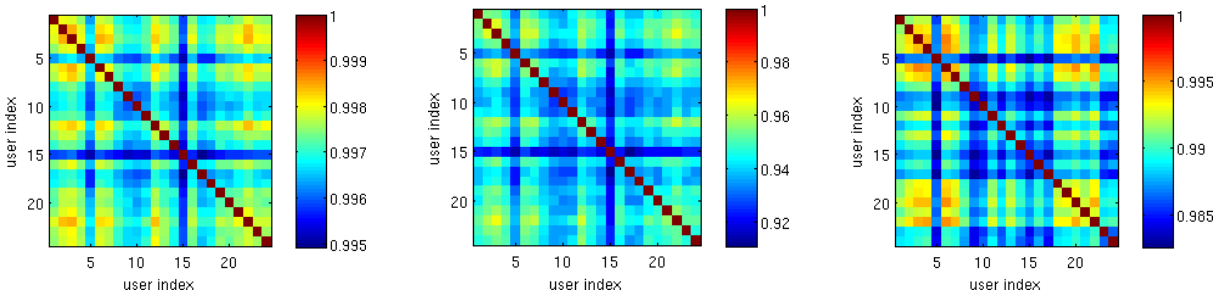

Figure 4: UNBC-MSPEAD dataset: similarity matrices of the learnt θ_i for each subtask, obtained with (a) our method, (b) individual "ideal" classifiers and (c) generic classifiers.

ductive SVM (TSVM) [12] and transfer learning-based methods: STM [5], Kernel Mean Matching (KMM) [9] and Domain Adaptation SVM (DASVM) [3]. The results are shown in Tables 1 and 2. The values concerning the performance of STM, SVM, KMM, TSVM and DA-SVM are reported from [5]. The parameters of our system (λ_E, ρ and ϵ) are computed using an inner cross-validation loop on the source subjects.

Our method outperforms all the other approaches considering both the AUC and the F_1 Score. Specifically, the best performances are obtained using the EMD kernel, while with the DE kernel and the Fisher kernel they are slightly inferior. Among competing methods, TSVM performs poorly. We argue that this is due to the fact that all the source samples are retained in the training process. This may bias the classifier because training samples from irrelevant subjects can be included. KMM, DASVM and STM, instead, re-weight and possibly retain only significant samples, according to different criteria. In particular, only STM outperforms a generic classifier, underlying the importance of an appropriate re-weighting scheme. However, our approach achieves even better performance than STM. We believe this is due to the fact that STM represents the discrepancy between source and target distributions by the distance of the associated centroids. Conversely, the kernel functions of TPT better capture the similarity between data distributions.

4.2 Pain Expression Recognition

4.2.1 Dataset

The **UNBC-McMaster Shoulder Pain Expression Archive Database**[3] (UNBC-MSPEAD) [18] is composed of 200 video sequences of patients with shoulder injuries. It depicts 25 patients performing a series of active and passive range-of-motion tests with either their affected limb or the unaffected one. The dataset is annotated on a frame basis (48398 frames are labeled by experts using the Prkachin and Solomon Pain Intensity, PSPI, metric system [23]). Some frames are shown in Fig. 3.

4.2.2 Feature Extraction

To extract features from video sequences we follow the approach proposed in [4]. For each frame we use the eye locations provided in the UNBC-MSPEAD database to crop and warp the face region into a 128×128 pixel image. Then the resulting face image is divided into 8×8 blocks and Local Binary Pattern Histograms features [1] are extracted on each block. Following the pipeline reported in [4] we adopt *uniform* $LBP^{u2}_{8,1}$, where *u2* means "uniform" and $(8, 1)$ represents 8 sampling points on a circle of radius 1. The resulting 59-dimensional feature vectors for each block are concatenated resulting into a descriptor of $8 \times 8 \times 59 = 3776$ dimensions. Finally, Principal Component Analysis is applied to reduce feature dimensions retaining 90% of the variance. The dimension of the final feature vectors is 334.

4.2.3 Results

Following [4], our experiments are conducted using a leave-one-subject-out evaluation scheme. However, since there is no pain exhibited in the videos of one subject (*i.e.* PSPI score is equal to 0 for all the frames), we exclude them from the experiments. Hence, the final number of subjects considered, both at training and at testing time, is 24. As in [4], every fame is tested independently of the others (no temporal information is used) and with a binary output (pain/no pain).

In order to allow a comparison of our results with [4], we evaluate the performance using AUC and we compare against: a generic classifier (SVM) trained using only the source samples (no domain adaptation), Transductive Transfer Adaboost (TTA) [4], Transductive SVM (TSVM) [12] and Selective Transfer Machine (STM) [5]. For TTA, we report the performances published by Chen *et al.* in [4], while for the last two algorithms, we use the codes publicly available[4,5]. The parameters of our system (λ_E, ρ and ϵ) are computed using an inner cross-validation loop on the source subjects. The results are shown in Fig. 5 and Fig. 6. Figure 5 compares different kernel choices in our approach. As in the case of the CK+ dataset, the best performances are obtained using the EMD kernel. In the rest of the experiments, we refer as TPT our method with the EMD kernel.

Figure 6 shows the accuracy of our approach with respect to the baseline methods. TPT outperforms all the other algorithms. However, the improvement is modest with respect to what observed on the CK+ dataset (see Sec. 4.1.3). In general, for this dataset, personalization does not provide much benefits with respect to a generic classifier (SVM). We attribute this finding to the following two main reasons. Firstly, this dataset is much more difficult than the CK+: while in the latter all the faces have a frontal pose, in UNBC-MSPEAD there are large pan and pitch rotations, expressions are spontaneous and inter-individual differences are pronounced. Moreover, in the CK+, only the emotion peaks are annotated (*i.e.* the last frame of each video), while in UNBC-MSPEAD all the frames are labeled, and the dif-

[3]http://www.pitt.edu/~emotion/um-spread.htm

[4]http://svmlight.joachims.org/

[5]http://humansensing.cs.cmu.edu/software.html

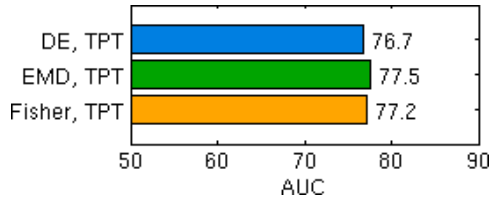

Figure 5: UNBC-MSPEAD dataset. Performances obtained with our method with different kernels.

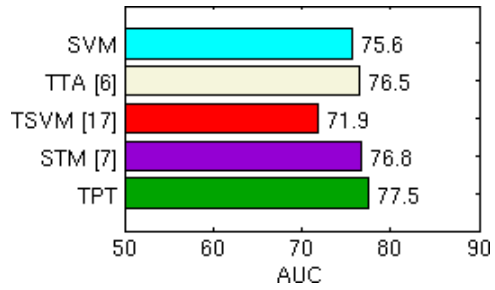

Figure 6: UNBC-MSPEAD dataset. Performances obtained with our method and baselines.

ference between pain and non pain expressions is more subtle. In fact, the pain intensity of positive samples varies from 1 to 16 and these samples are considered all equally positive. Sample frames of pain expressions at varying intensity levels are shown in Fig. 8. The second reason may be found in the fact that in the UNBC-MSPEAD dataset the number of subjects (and thus the number of individual classifiers $\boldsymbol{\theta}_i$) is lower than those in CK+ (24 vs. about 80-90, depending on the AU). In Sec. 4.3 we show how the accuracy of our approach varies as a function of N. Finally, comparing TPT with SVM on a subject basis (i.e., counting the number of persons for which there is an improvement of TPT over SVM), we observe that TPT achieves a higher AUC than SVM on 18 out of 24 subjects.

In order to further validate our approach, in Fig. 4 we show the similarity matrix $\mathbf{S}$ of the classifiers computed for each of the 24 subjects of the UNBC-MSPEAD dataset, defined as: $\mathbf{S}_{ij} = e^{-||\boldsymbol{\theta}_i - \boldsymbol{\theta}_j||_2}$, with $i, j = 1, ..., N$. In the leftmost matrix, each $\boldsymbol{\theta}_i$ is the classifier vector obtained with our method. In the center matrix, $\boldsymbol{\theta}_i$ is the "ideal" classifier for the i-th subject, *i.e.*, an SVM classifier computed using only the samples of the i-th individual, in the "ideal" hypothesis of having labeled samples for this target user. For sake of comparison, we also report the classifiers computed with a generic SVM using a leave-one-user-out protocol, in other words, using all the training samples excluding the ones of the i-th subject. The similarity matrix of the $\boldsymbol{\theta}_i$ learned with generic SVM are shown in the rightmost matrix. It is interesting to notice that the structure of the similarity matrix obtained with our method is very similar to the structure of the ideal classifier matrix, which visually confirms that our regression framework is effective in building personalized models.

Figure 8 shows the results obtained with our method on a sample video sequence. In details, it compares the values of the decision function obtained with TPT (*i.e.* $z = \mathbf{w}_t'\mathbf{x} + b_t$) and the original PSPI values (GT). GT and z have been normalized between 0 and 1 for ease of visual comparison. It is interesting to notice that the values of z are significantly aligned with the pain intensity values.

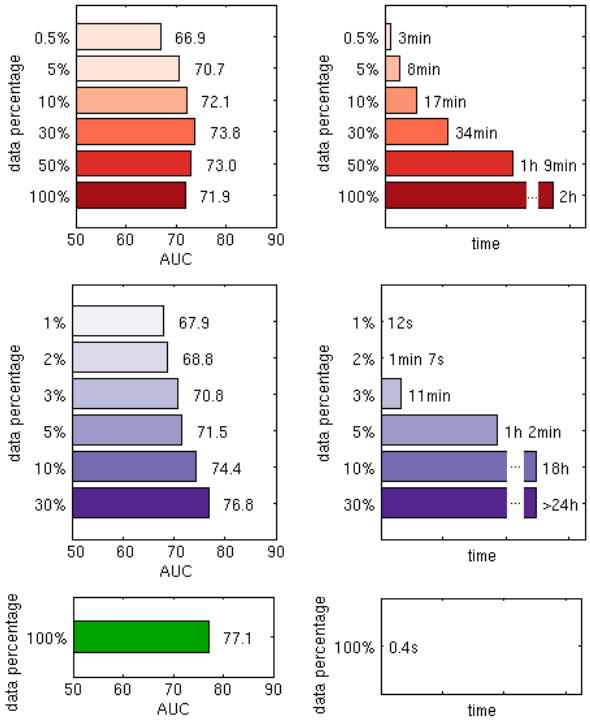

Figure 7: UNBC-MSPEAD dataset: (left) AUC and (right) average time for training a target classifier with different transfer learning methods: (top) TSVM [12], (center) STM [5] and (bottom) TPT. Results for [5, 12] are reported considering different percentage of source data samples.

The last part of this section is dedicated to compare TPT and state-of-the-art approaches with respect to the computational times. In Fig. 7 we report the performance and the training time of the methods for which code is available online, *i.e.* TSVM and STM. In [4] Chen *et al.* report the training time for TTA on UNBC-MSPEAD (17.6 minutes) but they do not mention the workstation they used, thus the results are not directly comparable. Our experiments were run on a 4 Cores 2.40GHz CPU machine.

The training times reported in Fig. 7 indicate the average times for training a single target classifier. All the methods but ours use all the source samples (suitably re-weighted) for training a target classifier, thus they require a much higher computational load. Conversely, in TPT training is split in three phases (Sec. 3), whose average time are the following: (i) computing domain-specific classifiers, ∼40 seconds, (ii) computing the kernel matrix, ∼4 seconds, (iii) computing the target-specific classifier using a pre-calculated kernel matrices, 0.4 seconds. It is worth noting that steps (i) and (ii) involve only source samples and are executed a single time. Only step (iii) is target-specific and it is the phase that realizes adaptation on a new subject. Thus in Fig. 7 we only report the computational times of step (iii). It is clear from these plots that both TSVM and STM do not scale well as the number of training samples increases. For instance, using only 10% of the source samples, TSVM needs 17 minutes on average for training an individual classifier, STM 18 hours and TPT only 0.4 seconds. Even if the accuracy of all the methods are comparable in this difficult dataset, our pro-

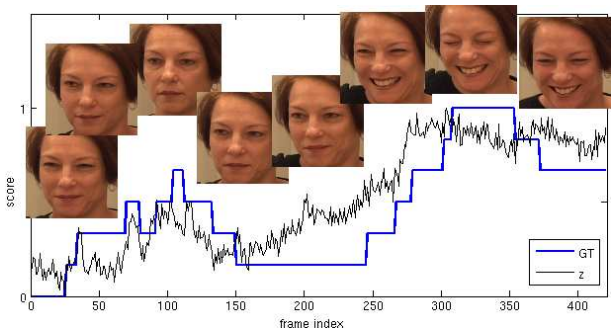

Figure 8: UNBC-MSPEAD dataset: ground truth labels (GT) compared to the value $z = \mathbf{w}_t'\mathbf{x}+b_t$ of the decision function obtained with TPT for one sample video sequence.

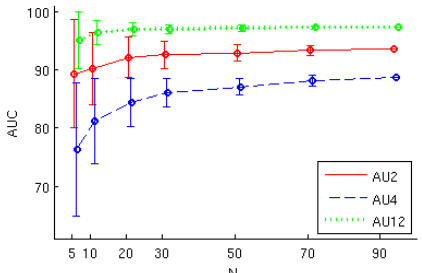

Figure 9: CK+ dataset. Performance of TPT at varying number of source users.

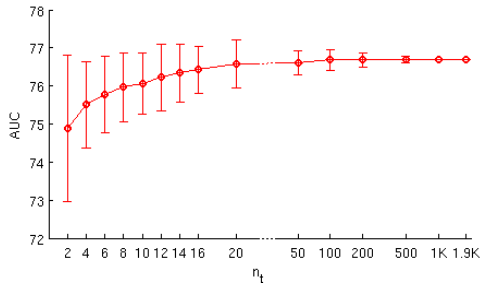

Figure 10: UNBC-MSPEAD dataset. Performance of TPT at varying number of target samples n_t.

posed approach largely outperforms all the other algorithms with respect to the computational cost. Moreover, independently from the kernel choice, we do not need to store the source samples. Indeed, the regression function value for a new target task is computed considering only on the kernel matrix. We believe that both memory storage and time efficiency are crucial in real world multimedia applications, where the construction of a personalized classifier needs to be fast and memory resources are usually limited (*e.g.*, with personal devices).

4.3 Relation between performance and number of source users and samples

In this section we analyse how the performance of our method depends on the number of source datasets N and target samples n_t. These experiments have been conducted respectively on the CK+, which contains the highest number of subjects - up to ∼90 (depending on the specific AU), while UNBC-MSPEAD dataset has only 25 subjects - and on the UNBC-MSPEAD, which contains a higher number of samples per subjects - around 1900 samples per user, while for CK+ it is around 150.

In the CK+ experiments, we randomly select a subset of $N+1$ users over all the subjects, varying N from 5 to ∼90. Then we test our method using a leave-one-user-out approach, in which in turn one user is chosen as the target subject and the remaining N are used as sources. For a given value of N, we repeat the experiments 100 times and the average values are reported in Fig. 9. The performances are measured computing the AUC. Error bars show the standard deviation from the mean. As expected, with a low number of source datasets the performance of TPT are rather modest, expecially for the AU which are more difficult to detect (*e.g.* AU4). In fact, the regression function $\widehat{f}(\cdot)$ (Sec. 3) is learned using N training samples (*i.e.* data distributions) and, if N is too small, it generalizes poorly.

We also analyse the impact of the number of target data points n_t on the recognition accuracy for the UNBC-MSPEAD dataset. Fig. 10 reports the average AUC obtained on 100 runs. On each run, a random subset of n_t points is selected for each target user. A leave-one-user-out protocol is used. When n_t increases the performance improves, as data distributions are better approximated when a large number of data points is employed.

5. CONCLUSIONS

In this paper we proposed a method for building person-specific facial expression classifiers as a way to deal with the inter-individual variability of the emotions. A personalized classifier for a new target subject is inferred without the need of acquiring labeled data. The proposed method is based on a regression framework which directly maps the unlabeled data distribution of a given person to the parameters of her/his personalized classifier. As far as we know, this is the first transductive parameter transfer approach in literature. Oppositely to previous transfer learning approaches operating in a transductive setting and based on instance reweighting the main advantage of our method is that we do not need to store and compare all the source and target samples. This leads to a significantly reduced computational cost. We empirically showed that our system achieves state-of-the-art accuracy on public benchmarks while being different orders of magnitude faster.

6. ACKNOWLEDGMENTS

This work was partially supported by the European 7th Framework Program, under grant VENTURI (FP7-288238) and xLiMe (FP7-611346) and by the cluster project Ageing at Home.

7. REFERENCES

[1] T. Ahonen, A. Hadid, and M. Pietikäinen. Face description with local binary patterns: Application to face recognition. *IEEE Trans. on PAMI*, 28(12):2037–2041, 2006.

[2] G. Blanchard, G. Lee, and C. Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *NIPS*, 2011.

[3] L. Bruzzone and M. Marconcini. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *IEEE Trans. on PAMI*, 32(5):770–787, 2010.

[4] J. Chen, X. Liu, P. Tu, and A. Aragones. Learning person-specific models for facial expression and action unit recognition. *Pattern Recognition Letters*, 34(15):1964–1970, 2013.

[5] W.-S. Chu, F. D. L. Torre, and J. F. Cohn. Selective transfer machine for personalized facial action unit detection. In *CVPR*, 2013.

[6] M. R. Daliri. Kernel earth mover's distance for eeg classification. *Clinical EEG and Neuroscience*, 44(3):182–187, 2013.

[7] H. Dibeklioğlu, T. Gevers, A. A. Salah, and R. Valenti. A smile can reveal your age: Enabling facial dynamics in age estimation. In *ACM Multimedia*, 2012.

[8] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *CVPR*, 2005.

[9] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

[10] T. Jaakkola, D. Haussler, et al. Exploiting generative models in discriminative classifiers. *NIPS*, 1999.

[11] T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *The Journal of Machine Learning Research*, 5:819–844, 2004.

[12] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, 1999.

[13] A. Kapoor, W. Burleson, and R. W. Picard. Automatic prediction of frustration. *International Journal of Man-Machine Studies*, 65(8):724–736, 2007.

[14] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012.

[15] S. Kumano, K. Otsuka, D. Mikami, and J. Yamato. Recognizing communicative facial expressions for discovering interpersonal emotions in group meetings. In *Proceedings of the 2009 International Conference on Multimodal Interfaces*, ICMI-MLMI '09, 2009.

[16] G. Littlewort, M. S. Bartlett, and K. Lee. Faces of pain: automated measurement of spontaneousallfacial expressions of genuine and posed pain. In *ICMI*, 2007.

[17] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *CVPR Workshops*, 2010.

[18] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, S. W. Chew, and I. Matthews. Painful monitoring: Automatic pain monitoring using the unbc-mcmaster shoulder pain expression archive database. *Image Vision Comput.*, 30(3):197–205, 2012.

[19] A. Martinez and S. Du. A model of the perception of facial expressions of emotion by humans: Research overview and perspectives. *Journal of Machine Learning Research*, 13:1589–1608, 2012.

[20] I. Mironica, J. Uijlings, N. Rostamzadeh, B. Ionescu, and N. Sebe. Time matters!: capturing variation in time in video using fisher kernels. In *ACM Multimedia*, 2013.

[21] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[22] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007.

[23] K. M. Prkachin and P. E. Solomon. The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain. *Pain*, 139(2):267–274, 2008.

[24] E. Ricci, G. Zen, N. Sebe, and S. Messelodi. A prototype learning framework using emd: Application to complex scenes analysis. *IEEE Trans. on PAMI*, 35(3):513–526, 2013.

[25] S. D. Roy, T. Mei, W. Zeng, and S. Li. Socialtransfer: cross-domain transfer learning from social streams for media applications. In *ACM Multimedia*, 2012.

[26] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[27] E. Sangineto. Pose and expression independent facial landmark localization using dense-surf and the hausdorff distance. *IEEE Trans. on PAMI*, 35(3):624–638, 2013.

[28] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2):200–215, 2011.

[29] K. Sikka, A. Dhall, and M. Bartlett. Weakly supervised pain localization using multiple instance learning. *IEEE FG*, 2013.

[30] M. Tkalcic, A. Odic, A. Kosir, and J. Tasic. Affective labeling in a content-based recommender system for images. *IEEE Trans. on Multimedia*, 15(2):391–400, Feb 2013.

[31] D. Tuia, J. Verrelst, L. Alonso, F. Perez-Cruz, and G. Camps-Valls. Multioutput support vector regression for remote sensing biophysical parameter estimation. *IEEE Geoscience and Remote Sensing Letters*, 8(4):804–808, July 2011.

[32] M. F. Valstar, B. Jiang, M. Mehu, M. Pantic, and K. Scherer. The first facial expression recognition and analysis challenge. In *IEEE FG*, 2011.

[33] M. F. Valstar, M. Pantic, Z. Ambadar, and J. F. Cohn. Spontaneous vs. posed facial behavior: automatic analysis of brow actions. In *ICMI*, 2006.

[34] A. Vinciarelli, M. Pantic, and H. Bourlard. Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 27(12):1743–1759, 2009.

[35] Y. Yang, Y. Yang, Z. Huang, J. Liu, and Z. Ma. Robust cross-media transfer for visual event detection. In *ACM Multimedia*, 2012.

[36] M. Yeasin, B. Bullot, and R. Sharma. Recognition of facial expressions and measurement of levels of interest from video. *IEEE Trans. on Multimedia*, 8(3):500–508, 2006.

[37] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Trans. on PAMI*, 31(1):39–58, 2009.