Selective Transfer Machine for Personalized Facial Action Unit Detection

Wen-Sheng Chu[†] **Fernando De la Torre**[†] **Jeffery F. Cohn**[†‡]
[†]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213
[‡]Department of Psychology, University of Pittsburgh, Pittsburgh, PA 15260

Abstract

Automatic facial action unit (AFA) detection from video is a long-standing problem in facial expression analysis. Most approaches emphasize choices of features and classifiers. They neglect individual differences in target persons. People vary markedly in facial morphology (e.g., heavy versus delicate brows, smooth versus deeply etched wrinkles) and behavior. Individual differences can dramatically influence how well generic classifiers generalize to previously unseen persons. While a possible solution would be to train person-specific classifiers, that often is neither feasible nor theoretically compelling. The alternative that we propose is to personalize *a generic classifier in an unsupervised manner (no additional labels for the test subjects are required). We introduce a transductive learning method, which we refer to Selective Transfer Machine (STM), to personalize a generic classifier by attenuating person-specific biases. STM achieves this effect by simultaneously learning a classifier and re-weighting the training samples that are most relevant to the test subject. To evaluate the effectiveness of STM, we compared STM to generic classifiers and to cross-domain learning methods in three major databases: CK+ [20], GEMEP-FERA [32] and RU-FACS [2]. STM outperformed generic classifiers in all.*

1. Introduction

The face is one of the most powerful channels of nonverbal communication. Facial expression provides cues about emotion, intention, alertness, pain, and personality, regulates interpersonal behavior, and communicates psychiatric and biomedical status among other functions. The Facial Action Coding System (FACS) [14] is the most comprehensive, anatomically-based system for encoding expression. FACS segments the visible effects of facial muscle activation into "action units" (AUs). Each AU is related to one or more facial muscles. FACS describes facial activity on the basis of 33 unique action units (AUs), as well as several categories of head and eye positions and other movements. Facial movement is thus described in terms of constituent components, or AUs.

Automatic facial action unit detection (AFA) confronts a series of challenges. These include changes in pose, scale, illumination, occlusion, and individual differences in face shape, texture, and behavior. Face shape and texture differ between and within sexes and ethnic and racial backgrounds, differ with age and exposure to the elements, and differ in rates of behavior. Some people smile broadly and frequently; others rarely or with smile controls, which counteract the upward pull of the zygomatic major on the lip corners. These and other sources of variation represent considerable challenges for computer vision. Then there is the challenge of automatically detecting facial actions that require significant training and expertise even for human coders, as has been recently reported in the first Facial Expression Recognition and Analysis Challenge [32].

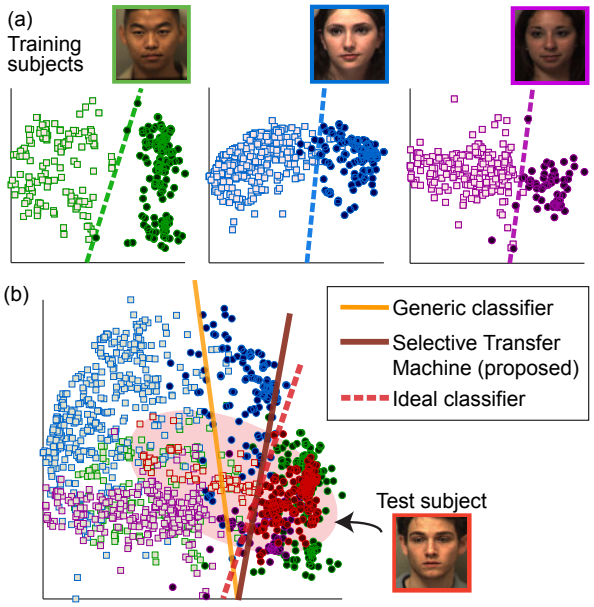


Figure 1. (a) 2D PCA projections of positive and negative samples for AU12 (lip-corner raiser). While ideal classifiers, trained and tested separately for each subject, correctly separate positive and negative samples (denoted by squares and circles, respectively) for each subject, (b) generic classifiers trained on data from all 3 subjects generalize poorly when applied to a previously unknown subject. Selective transfer machine, which personalizes the generic classifier, reliably separates AU12 for the unseen subject.

To address these issues, previous work has focused on identifying optimal feature representations and classifiers. See [11, 21, 32] for a review. While improvements have been achieved, generalizability of classifiers to previously unseen persons remains a continuing challenge. Fig. 1(a) illustrates an example of how a simple linear classifier can separate the positive samples of AU12 (obliquely raised lip corners, seen in smiling) from negative samples (*i.e.*, all other AUs). In this case, we use all available data of the same subject for training, and we call this the *ideal classifier*. However, when a classifier is learned using training data from all subjects (Fig. 1(b)) and tested on a subject excluded from the training set, it fails to generalize well. When a classifier is trained on all available subjects, it is referred as *generic*. We propose that impaired generalizability occurs in part because of individual differences among subjects. In the example shown, these differences include sex, skin color, and illumination. Our guiding hypothesis is that these factors lead generic classifiers to perform better or worse on some subjects than others.

To mitigate the person-specific biases, this paper explores the idea of *personalizing* a generic classifier. Generic classifiers are personalized using no AU labels from test subjects. We propose a new transductive technique called Selective Transfer Machine (STM). STM personalizes the generic classifier in an unsupervised manner to compensate for person-specific biases, and greatly improves generalizability, see Fig. 1(b). We illustrate the benefits of our approach in the task of facial AU detection in three major datasets of posed and spontaneous facial expressions. To the best of our knowledge, this is the first work to investigate personalizing a classifier for facial expression analysis.

2. Related work

Related work includes AU detection and cross-domain adaptation. We briefly review each in turn.

2.1. Facial AU detection

Automatic detection of AUs entails at least three steps. These are tracking and registration, feature extraction and possible data reduction, and classifier selection.

Tracking non-rigid facial features has been a long standing problem in computer vision. Most popular approaches to non-rigid tracking have been Active Appearance Models [22] or more recent advances such as Constrained Local Models [25] or discriminative AAMs [35]. It is beyond the scope of this paper to review all of them but we refer the reader to recent papers on this topic [25, 35].

Once the tracking is done and the face is registered, many features have been proposed to use for AU detection such as pixel intensities, edges, SIFT [39], DAISY [39], Gabor jets [2], compositional features [37] and many others, but as shown in the first facial expression recognition challenge [32], none of them has clearly been shown to be superior to one another.

Two main approaches have been pursued for designing classifiers for AU detection. One is static modeling, which is typically posed as a discriminative classification problem in which each video frame is evaluated independently [2]. The other is temporal modeling in which frames are segmented into sequences and modeled with a variant of Dynamic Bayesian Networks (*e.g.*, Hidden Markov Models, Conditional Random Fields) [7, 26, 33]. For instance, Tong *et al*. [30] used Dynamic Bayesian Networks with appearance features to model AU co-occurrence. Alternatively, Simon *et al*. [27] proposed to use a structural-output SVM for detecting the starting and ending frames of each AU. Recently, Rudovic *et al*. [24] considered the ordinal information in a Conditional Random Field to model the relations between temporal segments.

Interested readers may refer to [11, 21, 32] for more complete surveys of AU detection methods. Common to all of these approaches is the assumption that training and test data come from the same distribution. STM makes no such assumption. It therefore seeks to personalize the classifier by automatically re-weighting training samples that are most relevant to each test subject.

2.2. Cross-domain adaptation

Our approach is motivated by increased concern about database imbalance and bias in computer vision. In real-world data, labels of interest often occur infrequently, and features can vary markedly between and within datasets. Torralba and Efros [31] discovered significant biases in object categorization; as a remedy, they encouraged advances in *domain adaptation* to cope with dataset biases. Aytar and Zisserman [1] proposed to transfer pre-learned models to regularize the training of a new object class. Recently, Khosla *et al*. [18] combined a specific and a common discriminative model across several tasks to remove bias. These techniques used a supervised approach to learning in which one or more labeled instances are required from the target domain. They cannot be applied to new domains or subjects when one has no prior knowledge of them. In contrast, our approach is fully unsupervised, uses no labeled instances, and hence well suited to the problem of generalizing learning to new domains or new subjects in our case.

Close to our approach is a special case in unsupervised domain adaptation known as *covariate shift* [28], where training and test domains follow different distributions but the label distributions remain the same. Dudík *et al*. [13] inferred the re-sampling weights through maximum entropy density estimation. Maximum Mean Discrepancy (MMD) [3] measured the discrepancy between two different distributions in terms of expectations of empirical samples. Without estimating densities, Transductive SVM (T-SVM)

[17] simultaneously learn a decision boundary and maximize the margin in the presence of unlabeled patterns. Domain adaptation SVM [5] extends T-SVM by progressively adapting the discriminant function to the target domain. SVM-KNN [38] labels a single query using an SVM trained on its k neighborhood of the training data. Each of these methods uses either all or a subset of the training data. Unlike previous approaches, STM learns weights on individual training instances and hence makes better use of the data.

Considering distribution mismatch, Kernel Mean Matching (KMM) [16] directly infers the re-sampling weights by matching training and test distributions. Following this idea, Yamada *et al*. [36] estimated the relative importance weight and learn from weighted training samples for 3D human pose estimation. Interested reader may refer to [23] for a complete review. However, these methods take a two-step approach that first estimates the sampling weights and then trains a re-weighted classifier/regressor. On the contrary, STM jointly optimize the weights as well as the classifier parameters, and hence preserves discriminant property of the new decision boundary. In Sec. 4 the benefit of STM over KMM will become more apparent.

3. Selective Transfer Machine (STM)

This section describes the proposed STM approach for personalizing a generic classifier. Unlike previous cross-domain methods [1, 12, 18], STM will not require labels from a test subject. We will use Support Vector Machine (SVM) as the classifier because it has been a popular classification tool for AU detection [9, 27, 34]. However, STM is not a classifier-dependent technique, and hence can be used with any classifier.

Problem formulation: The main idea behind the STM is to re-weight more the training samples that are closer to the test samples. The classifiers trained on the re-weighted training samples will be more likely to fit the test subject. Let us denote the training set as $\mathcal{D}^{\text{tr}} = \{\mathbf{x}_i, y_i\}_{i=1}^{n_{\text{tr}}}, y_i \in \{+1, -1\}$ (see notation[1]). For notational simplicity, we stack 1 in each data vector $\mathbf{x}_i$ to compensate for the offset, *i.e.*, $\mathbf{x}_i \in \mathbb{R}^{d+1}$. We formulate STM as:

$$(\mathbf{w}, \mathbf{s}) = \arg\min_{\mathbf{w},\mathbf{s}} R_{\mathbf{w}}(\mathcal{D}^{\text{tr}}, \mathbf{s}) + \lambda \Omega_{\mathbf{s}}(\mathbf{X}^{\text{tr}}, \mathbf{X}^{\text{te}}), \quad (1)$$

where $R_{\mathbf{w}}(\mathcal{D}^{\text{tr}}, \mathbf{s})$ is the SVM empirical risk defined on training set $\mathcal{D}^{\text{tr}}$ with each instance weighted by $\mathbf{s} \in \mathbb{R}^{n_{\text{tr}}}$, *i.e.*, each entry s_i of $\mathbf{s}$ corresponds to a positive weight for the sample $\mathbf{x}_i$. $\Omega_{\mathbf{s}}(\mathbf{X}^{\text{tr}}, \mathbf{X}^{\text{te}})$ measures the distribution mismatch between the training and test distribution as a function of $\mathbf{s}$. The lower the value of $\Omega_{\mathbf{s}}$, the more similar the training and test distributions are. $\lambda \geq 0$ is a tradeoff to balance the risk and the distribution mismatch. The goal of the STM is to jointly optimize the penalized SVM $\mathbf{w}$ as well as the selective coefficient $\mathbf{s}$, such that the resulting personalized classifier can better remove person-specific biases.

Penalized SVM: The first term in STM, $R_{\mathbf{w}}(\mathcal{D}^{\text{tr}}, \mathbf{s})$, is the empirical risk of a penalized SVM, where each training instance is weighted by its relevance to the test data. The instance-wise weighted SVM minimizes:

$$R^{\text{lin}}_{\mathbf{w}}(\mathcal{D}^{\text{tr}}, \mathbf{s}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n_{\text{tr}}} s_i L_p(y_i, \mathbf{w}^\top \mathbf{x}_i), \quad (2)$$

where $L_p(y, \cdot) = \max(0, 1 - y\cdot)^p$, and p is either 1 (hinge loss) or 2 (quadratic loss), but generally, L could be any loss function. Eq. (2) can be extended to a nonlinear version by introducing a kernel matrix $\mathbf{K}_{ij} := k(\mathbf{x}_i^{\text{tr}}, \mathbf{x}_j^{\text{tr}})$ corresponding to a kernel function k induced from the nonlinear feature mapping $\varphi(\cdot)$. Using the representer theorem [8], the penalized SVM in (2) can be rewritten as:

$$R^{\text{nonlin}}_{\boldsymbol{\beta}}(\mathcal{D}^{\text{tr}}, \mathbf{s}) = \frac{1}{2}\boldsymbol{\beta}^\top \mathbf{K}\boldsymbol{\beta} + C\sum_{i=1}^{n_{\text{tr}}} s_i L_p(y_i, \mathbf{K}_i^\top \boldsymbol{\beta}). \quad (3)$$

Unlike most existing work, we will train the kernel SVM in the primal due to its simplicity and efficiency using the Newton's method that has quadratic convergence [8]. In addition, standard packages that solve the SVM in the primal do not incorporate instance-wise weights. Details of the optimization will be given in Sec. 4.

Domain mismatch: The second term in STM, $\Omega_{\mathbf{s}}(\mathbf{X}^{\text{tr}}, \mathbf{X}^{\text{te}})$, is the domain mismatch, and it has the objective to find a re-weighting function for minimizing the mismatch between training and test domains. In previous cross-domain learning methods, the re-weighting function may be computed by separately estimating the densities and then the weights (*e.g.*, [29]). However, this two-step strategy is not practical and increases the estimation error while taking the ratio of estimated densities [29].

An intuitive way to reassign weights is to compute the ratio between training and test densities. However, such densities are unavailable in real world applications. Here we adopt the Kernel Mean Matching (KMM) [16] method to reduce the difference between the means of the training and test distributions in the Reproducing Kernel Hilbert Space $\mathcal{H}$. KMM computes the instance-wise re-weighting s_i that minimizes

$$\Omega_{\mathbf{s}}(\mathbf{X}^{\text{tr}}, \mathbf{X}^{\text{te}}) = \left\| \frac{1}{n_{\text{tr}}}\sum_{i=1}^{n_{\text{tr}}} s_i \varphi(\mathbf{x}_i^{\text{tr}}) - \frac{1}{n_{\text{te}}}\sum_{j=1}^{n_{\text{te}}} \varphi(\mathbf{x}_j^{\text{te}}) \right\|^2_{\mathcal{H}}. \quad (4)$$

Introducing $\kappa_i := \frac{n_{\text{tr}}}{n_{\text{te}}}\sum_{j=1}^{n_{\text{te}}} k(\mathbf{x}_i^{\text{tr}}, \mathbf{x}_j^{\text{te}})$, $i = 1, \ldots, n_{\text{tr}}$, that captures the closeness between training and each test sample, finding a suitable $\mathbf{s}$ in (4) can be rewritten as a quadratic

[1] Bold capital letters $\mathbf{X}$ denote a matrix; $\mathbf{X}_i$ represents the i^{th} column of the matrix $\mathbf{X}$. Bold lower-case letters a column vector $\mathbf{x}$; x_j denotes the scalar in the j^{th} element of $\mathbf{x}$. All non-bold letters represent scalars. $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ is an identity matrix.

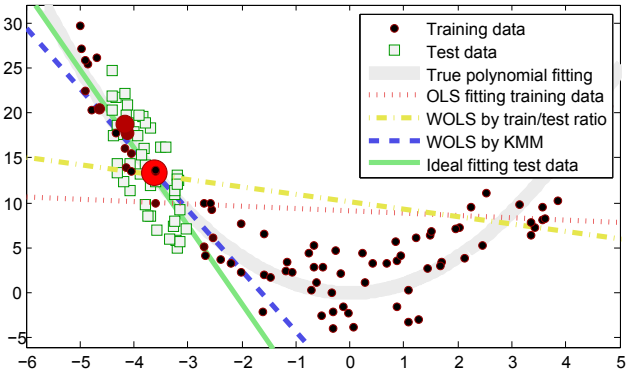


Figure 2. Fitting a line to a quadratic function using KMM and other re-weighting approaches. Circles represent the training data and squares the test data. The larger size (or more red) of training data, the more weight KMM adapted for fitting. As can be observed, KMM put higher weights in the training samples closer to the test samples. Compared to standard OLS or WOLS, this allows to better approximate linear models (lines) for the test data.

programming (QP):

$$\min_{\mathbf{s}} \ \frac{1}{2}\mathbf{s}^\top \mathbf{K}\mathbf{s} - \boldsymbol{\kappa}^\top \mathbf{s}, \tag{5}$$
$$\text{s.t.} \ \ s_i \in [0, B], \left|\sum_{i=1}^{n_{\text{tr}}} s_i - n_{\text{tr}}\right| \leq n_{\text{tr}}\epsilon.$$

where B in the first constraint defines a scope bounding discrepancy between the training/test distributions P_{tr} and P_{te}. For $B \to 1$, we obtain the unweighted solution. The latter constraint ensures that the resulting measure $\mathbf{s}(x)P_{tr}(x)$ is close to a probability distribution according to Hoeffding's inequality [16] . Large values of κ_i indicate important observations $\mathbf{x}_i^{\text{tr}}$ and are likely to lead to large s_i. A major advantage of KMM is that it does not require the estimation of biasing densities or selection probabilities. Fig. 2 illustrates its effect on a synthetic data. As shown, KMM can predict the ideal fitting well, while standard Ordinary Least Square (OLS) and Weighted OLS (WOLS) with training/test ratio fail to predict the true test fitting.

Similarities and differences between STM and cross-domain learning methods: Both STM and cross-domain learning methods seek to compensate for data-specific biases. They differ in how they accomplish this. We compare and contrast STM with three widely-used cross-domain learning approaches: T-SVM [17], KMM [16] and DA-SVM [5]. T-SVM [17] equally weights all the training data; by contrast, STM gives greater weight to training data that are more relevant to a given test subject. T-SVM is formulated as integer programming, which is difficult to optimize and scale to large problems. On the other hand, STM is formulated as a biconvex problem and therefore assures convergence. Both KMM [16] and STM re-weight the data. KMM does re-weighing only once, while STM does so in an iterative manner. STM uses the outcomes of training to refine the weighting at successive steps. In this way, STM is able to correct sub-optimal weights. From this perspective, KMM can be viewed as a special case of STM (see Sec. 4 for more discussions) in which re-weighting is performed only at an initial step. DA-SVM [5], similar to T-SVM, learns a classifier without re-weighting the training data. STM, as noted, reassigns weights in light of successive outcomes. A further difference is that DA-SVM may fail to converge, while STM always converges.

Algorithm 1: Selective Transfer Machine

Input : $\mathbf{X}^{\text{tr}}$, $\mathbf{X}^{\text{te}}$, parameters C, λ
Output: Classifier $\mathbf{w}$ and instance-wise weights $\mathbf{s}$
Initialize training loss $\boldsymbol{\ell}_p \leftarrow \mathbf{0}$;
while *not converged* **do**
  Find the instance-wise re-weighting $\mathbf{s}$ by solving the QP in (6);
  Find the classifier $\mathbf{w}$ by solving the penalized SVM in (2) or (3);

4. Optimization for STM

To minimize Eq. (1) we adopt the Alternate Convex Search method [15] that alternates between solving two convex subproblems over the hyperplane $\mathbf{w}$ and the selective coefficient $\mathbf{s}$. As the STM objective in (1) is biconvex, that is, convex in $\mathbf{w}$ when $\mathbf{s}$ is fixed (quadratic in $\mathbf{w}$ and L_p is convex), and convex in $\mathbf{s}$ when $\mathbf{w}$ is fixed (since $\mathbf{K} \succeq 0$). Under these conditions, the alternated optimization approach is guaranteed to monotonically decrease the objective function. Because the function is bounded below, it will converge to a critical point. Algorithm 1 summarizes the STM algorithm.

Minimizing over s: Denote the training losses as $\ell_p := L_p(y_i, \mathbf{w}^\top \mathbf{x}_i)$, $i = 1, \ldots, n_{\text{tr}}$. The optimization over $\mathbf{s}$ can be rewritten into the following QP:

$$\min_{\mathbf{s}} \ \frac{1}{2}\mathbf{s}^\top \mathbf{K}\mathbf{s} + (\frac{C}{\lambda}\boldsymbol{\ell}_p - \boldsymbol{\kappa})^\top \mathbf{s} \tag{6}$$
$$\text{s.t.} \ \ 0 \leq s_i \leq B, n_{\text{tr}}(1-\epsilon) \leq \sum_{i=1}^{n_{\text{tr}}} s_i \leq n_{\text{tr}}(1+\epsilon).$$

This can be solved efficiently using interior point methods or other successive optimization procedure such as Alternating Direction Method of Multipliers (ADMM) [4]. Since $\mathbf{K} \succeq 0$ by definition, the QP has only one global optima. To make the algorithm numerically stable, it is possible to add a small ridge σ on the diagonal, *i.e.*, $\mathbf{K} = \mathbf{K} + \sigma\mathbf{I}_n$ ($\sigma = 10^{-8}$ in our case).

Note that the procedure here is different from the original KMM as in each iteration the weighting will be refined

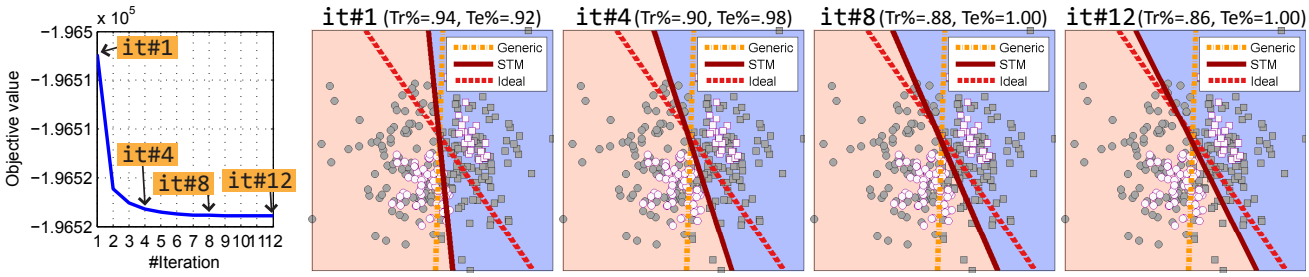


Figure 3. Comparison of a generic SVM, personalized STM, and ideal classifier for synthetic data. The left most figure shows the convergence curve of the objective value where STM converges in 12 iterations. Figures `it#1,4,8,12` with training/test accuracy (Tr% and Te%) show the hyperplanes in corresponding iterations, where grey (shaded) dots denote training data and white (unshaded) dots denote test data; circle/square patterns denote positive/negative classes respectively. Note that `it#1` indicates the result of KMM [16]. STM improves separation relative to generic SVM as early as the first iteration and converges close to the ideal hyperplane by the 12-th iteration.

through the training loss ℓ_p given by the penalized SVM. Since KMM is an unsupervised approach and does not use the label information, it is possible that the selected samples are noisy. Introducing the training loss helps preserve the discriminant property of the new decision boundary, and hence leads to a personalized classifier that is close to the ideal one. This effect can be also observed from minimizing the linear term in (6), where the instances with greater loss will be given smaller weights. From this perspective, the standard two-step KMM can be regarded as a special case as the first iteration of STM.

Fig. 3 illustrates the iterative effect on a synthetic example for learning a target-specific classifier. As shown in `it#1`, KMM fails to approach the ideal hyperplane since it does not impose any constraint in the classification performance. On the other hand, STM simultaneously considers training loss and the weightings, and thus encourages the training samples close to the test samples be well classified. As can be observed in Fig. 3, as the iterations proceed, the STM separation hyperplane approaches toward the ideal one for the target data.

Minimizing over w: In the case of training loss ℓ_2 being quadratic, the gradient and Hessian of the penalized linear SVM in (2) can be written as:

$$\nabla_{\text{lin}} = \mathbf{w} + 2C \sum_{i \in sv} s_i (\mathbf{w}^\top \mathbf{x}_i - y_i)\mathbf{x}_i, \tag{7}$$

$$H_{\text{lin}} = \mathbf{I}_d + 2C \sum_{i \in sv} s_i \mathbf{x}_i \mathbf{x}_i^\top, \tag{8}$$

where sv denotes the index set of support vectors. Let us denote $\mathbf{S} = \text{diag}(\mathbf{s}) \in \mathbb{R}^{n \times n}$ the re-weighting matrix, $\mathbf{y} \in \mathbb{R}^n$ the label vector, and $\mathbf{I}^0 \in \mathbb{R}^{n \times n}$ the proximity identity matrix with the first n_{sv} diagonal elements being 1 and the others 0. We can derive the gradient w.r.t. the expansion coefficient $\boldsymbol{\beta}$ for the penalized nonlinear SVM in (3) as:

$$\nabla_{\text{nonlin}} = \mathbf{K}\boldsymbol{\beta} + 2C\mathbf{K}\mathbf{S}\mathbf{I}^0(\mathbf{K}\boldsymbol{\beta} - \mathbf{y}), \tag{9}$$

$$H_{\text{nonlin}} = \mathbf{K} + 2C\mathbf{K}\mathbf{S}\mathbf{I}^0\mathbf{K}. \tag{10}$$

Given the gradients and Hessians, the penalized SVMs can be solved through standard Newton's methods or conjugate gradient. In the case of using the ℓ_1 hinge loss that is not differentiable, one can use subgradient methods or a differentiable approximation of the Huber loss [8].

5. Experiments

STM was compared for AU detection with generic SVM and cross-domain learning approaches in three widely used databases that vary in duration, extent of out-of-plane head motion, and spontaneity of facial expression. The 8 most frequently occurring AUs across the databases were selected for analysis.

5.1. Datasets

1) Extended Cohn-Kanade (CK+) [20] contains image sequences of posed and non-posed spontaneous expressions with frontal pose. Image sequences average about 20 frames in length; they begin with neutral expression and proceed to a peak, which is AU-labelled. We used 593 posed images sequences from 123 subjects.

2) GEMEP-FERA [32] is a subset of the GEMEP corpus. Head pose is frontal. Trained actors portray 18 emotions. We used the training subset of 87 videos from 7 actors, which ranged in length between 40 and 110 frames.

3) RU-FACS [2] consists of recorded interviews of 100 young adults. Interviews were approximately 2.5 minutes in duration. Head pose was frontal with small to moderate out-of-plane rotation. We had access to 34 of the interviews, of which video from 5 subjects could not be processed for technical reasons (*e.g.*, noisy video). Thus, the experiments reported here were conducted with data from 29 participants with more than 180,000 frames in total.

5.2. Experimental settings

Face tracking/alignment: 66 landmarks in the face were tracked using person-specific Active Appearance Models (AAMs) [22].

Feature extraction: Appearance features were extracted as SIFT descriptors [39]. Because AUs are localized

Table 1. Comparison between STM and PS classifiers

	AUC			$F1$ Score		
AU	PS_1-SVM	PS_2-SVM	STM	PS_1-SVM	PS_2-SVM	STM
1	48.0	72.4	**79.2**	45.0	54.8	**61.9**
2	46.5	71.1	**80.2**	45.9	55.7	**64.3**
4	62.6	61.9	**66.5**	46.6	40.7	**60.4**
6	70.3	80.0	**86.4**	60.2	69.7	**78.5**
7	47.5	54.3	**72.4**	49.4	55.3	**58.4**
12	65.7	74.0	**72.3**	69.5	70.4	**72.6**
15	41.4	64.0	**70.5**	44.5	49.0	**56.0**
17	32.6	**70.3**	61.7	25.0	**40.3**	36.3
Avg	51.8	68.5	**73.6**	48.3	54.5	**61.0**

Table 2. Comparisons on the CK+ dataset [20]

	AUC					$F1$ Score				
AU	SVM	KMM	TSVM	DASVM	STM	SVM	KMM	TSVM	DASVM	STM
1	79.8	68.9	69.9	72.6	**88.9**	61.1	44.9	56.8	57.7	**62.2**
2	**90.8**	73.5	69.3	71.0	87.5	73.5	50.8	59.8	64.3	**76.2**
4	74.8	62.2	63.4	69.9	**81.1**	62.7	52.3	51.9	57.7	**69.1**
6	89.7	87.7	60.5	**94.7**	94.0	75.5	70.1	47.8	68.2	**79.6**
7	82.1	68.2	55.7	61.4	**91.6**	59.6	47.0	43.8	53.1	**79.1**
12	88.1	89.5	76.0	**95.5**	92.8	76.7	74.5	59.6	59.0	**77.2**
15	93.5	66.8	49.9	94.1	**98.2**	75.3	44.4	40.4	76.9	**84.8**
17	90.3	66.6	73.1	94.7	**96.0**	76.0	53.2	61.7	81.4	**84.3**
Avg	86.1	72.9	64.7	81.7	**91.3**	70.0	54.7	52.7	64.8	**76.6**

to specific face regions, descriptors were computed within 36×36 pixel regions at predetermined facial landmarks (9 for the upper face and 7 for the lower face). Dimensionality was reduced using PCA, retaining 98% of energy .

AU selection & evaluation: The 8 most frequently occurring AU across the databases were selected for analysis. Positive samples were frames in which a given AU was present, and negative samples in which it was not. To provide a more objective evaluation, we report both Area Under the ROC Curve (AUC) and $F1$ score, which is defined as $F1 = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}$. Both metrics are widely used in the literature and convey non-redundant information. AUC quantifies the relation between true and false positives. $F1$ quantifies the trade-off between recall and precision.

5.3. Comparison with person-specific classifiers

A natural comparison with STM is a person-specific (PS) classifier. PS can be defined in at least two ways. One, which is more common, is a classifier that has been trained and tested on the same subject. We refer to this usage as **PS_1**. The other meaning, which we refer to as **PS_2** or quasi-PS, is a classifier that has been tested on a subject that was included among others in a training set. For instance, consider the case in which data from five subjects are randomly assigned to training and testing sets. A **PS_2** classifier is trained and then tested on the test set. Thus each subject had data in both train and test sets. The GEMEP-FERA [32] defined PS in this way. For SVM, we evaluated PS both ways. For STM, only **PS_2** was possible.

We trained person-specific SVMs on both scenarios, and trained STM only on PS_2 to show the selection ability. It is not surprising that PS_2-SVM perform better than PS_1-SVM since PS_1-SVM was trained only on limited training data and thus suffers from overfitting. As PS_2-SVM was trained on all available subjects, it can be viewed as a generic classifier, as used in most literature on AU detection. However, as discussed in Sec. 1, generic classifiers could suffer from the biases and lead to suboptimal performance. On the other hand, STM consistently outperforms both person-specific classifiers since STM allows to select only relevant training data and fits better the test distribution. Fig. 4 shows the selection ratio of STM on initialization and after convergence using PS_2. Each row sums to 1 and each entry shows the portion of selected samples of training subjects with respect to each test subject. As shown in Fig. 4(b), when STM converges, it selects most of the training data that belongs to the target subject (higher diagonal values).

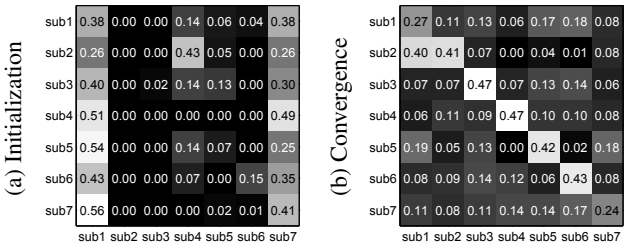


Figure 4. Selection ratio of STM for different subjects on the initialization and convergence step on PS_1. Each row sums to 1 and denotes a test subject. Each entry shows the portion of selected training samples w.r.t. each test subject.

5.4. Comparison with generic classifiers and domain adaptation approaches

This experiment compares the performance of STM against generic classifiers learned on the entire dataset, the covariate shift method KMM [16], a semi-supervised T-SVM [10], and the domain adaptation method DA-SVM [5]. We compared the methods on the CK+, RU-FACS and GEMEP-FERA databases. In this experiment, any sample of the test subjects is excluded from training.

We used the Gaussian kernel with a bandwidth that is the median distance between sample points. For KMM and STM we set $B = 1000$ so that none of s_i reached the upper bound, $\epsilon = \frac{\sqrt{n_{\text{tr}}}-1}{\sqrt{n_{\text{tr}}}}$, and cross-validated on the unweighted data (as suggested on [16]). For T-SVM we used the implementation in [10] since the original T-SVM [17] solves an integer programming and is not scalable to large size problem such as detecting AUs for a video of thousands of frames. We used a linear SVMs in all the methods. For the DA-SVM method we used LibSVM [6] and the penalized SVM discussed in Sec. 4 for the bound minimization problem with $\tau = 0.5$ and $\beta = 0.03$. Parameters for all methods were selected by cross-validation.

Table 3. Comparisons on the GEMEP-FERA dataset [32]

	AUC					$F1$ Score				
AU	SVM	KMM	T-SVM	DA-SVM	STM	SVM	KMM	T-SVM	DA-SVM	STM
1	71.5	43.3	72.2	83.3	**84.3**	56.5	48.5	60.3	59.1	**68.1**
2	73.9	51.0	74.3	**76.8**	73.3	56.9	50.2	58.5	57.1	**65.5**
4	58.5	53.5	42.8	**66.6**	60.0	43.5	39.8	36.9	**46.3**	43.3
6	80.4	60.2	81.1	**91.1**	87.7	63.7	58.7	63.8	**72.7**	71.6
7	66.9	59.4	70.8	**76.9**	75.4	63.1	63.5	63.7	**68.3**	66.2
12	77.7	58.8	74.8	74.5	**84.7**	79.1	68.4	77.6	75.5	**82.1**
15	55.5	58.7	67.2	67.5	**67.8**	33.4	35.2	35.2	**41.3**	39.3
17	59.8	51.8	63.8	**66.5**	63.3	32.0	27.8	36.2	**42.0**	35.9
Avg	68.0	54.6	68.4	**75.4**	74.5	53.5	49.0	54.0	57.8	**59.0**

Table 4. Comparisons on the RU-FACS dataset [2]

	AUC					$F1$ Score				
AU	SVM	KMM	T-SVM	DA-SVM	STM	SVM	KMM	T-SVM	DA-SVM	STM
1	72.0	74.0	72.0	77.0	**83.9**	40.8	37.7	37.4	35.5	**55.3**
2	66.6	58.6	71.1	76.5	**82.4**	35.7	32.2	36.2	34.1	**52.6**
4	74.8	62.2	50.0	76.4	**82.4**	25.2	14.5	11.2	**35.3**	30.4
6	89.1	88.8	61.6	60.3	**93.1**	58.3	39.2	33.1	42.9	**72.4**
12	86.7	87.0	86.7	84.4	**92.3**	61.9	63.0	62.6	71.4	**72.3**
14	71.8	67.8	74.4	70.4	**87.4**	31.3	25.8	25.8	40.9	**51.0**
15	72.5	68.8	73.5	58.1	**86.1**	32.3	29.5	32.3	34.9	**45.4**
17	78.5	76.7	79.5	75.7	**89.6**	39.5	35.6	44.0	46.5	**55.3**
Avg	76.5	72.3	71.1	72.3	**85.3**	40.6	37.3	40.6	42.7	**54.3**

Tables 2∼4 show the AUC and $F1$ scores on the CK+, GEMEP-FERA and RU-FACS databases. The linear SVM served as a baseline generic classifier. KMM failed to perform better than the baseline because it estimated the weights without using label information during the training. T-SVM performed similar to SVM in GEMEP-FERA and RU-FACS, but worse than SVM in CK+. This is because in CK+ the negative (neutral frames) and positive (peak frames) samples are more distinct compared to consecutive frames in GEMEP-FERA or RU-FACS. It is important to notice that for CK+, we used the last one third frames in the sequence to evaluate the generalization ability. Although the results are not directly comparable, STM achieved 91% AUC, which is slightly better than the best published results with 90.5% [19]. Note that by including the frames that are further away from the peak frames, the problem is more challenging for STM.

Unlike STM that used a penalized SVM, T-SVM did not consider re-weighting for training instances and make use of the losses for all training data. Hence it still suffer from person-specific biases, where irrelevant subjects still contribute equally during training. DA-SVM extends T-SVM by progressively labelling test patterns and removing labelled training patterns. Not surprisingly, DA-SVM shows better performance than KMM and T-SVM, because it used more relevant training samples and resulted in a better personalized classifier. However, similar to T-SVM, DA-SVM did not update the re-weightings using label information. Moreover it is not always guaranteed to converge to a correct solution. In our experiments, we faced the situation where DA-SVM failed to converge due to large amount of samples lying within the margin bounds. By contrast, STM is a biconvex formulation, and therefore guarantees to converge to a critical point and outperforms existing approaches. Observe that in Table 3, STM performed slightly worse in terms of AUC due to imbalanced data. However, using the $F1$ criterion, which is better suited for imbalanced detection task (as noted above), STM shows an improvement. In the larger RU-FACS dataset where more data is available, the improvement became clearer.

6. Conclusions

This paper proposed a transductive method to personalize a generic classifier for facial Action Unit (AU) detection. Our STM framework simultaneously learns the parameters of a classifier and the selective weights that minimizes the mismatch between the training and the test distributions. We show that STM translates to a biconvex problem, and propose a simple alternated minimization approach to optimize it in the primal. By attenuating the influence of inherent biases in morphology and behavior, we have shown that STM can achieve results that surpass non-personalized generic classifiers and approach the performance of classifiers that have been trained for individual persons (*i.e.*, person-dependent classifiers). The results have clearly demonstrated that STM outperforms existing classifiers when using the same protocol for training and testing. That is, STM proved comparable to cross-domain methods in the smaller CK+ and FERA databases. In the larger RU-FACS database, STM outperformed the cross-domain methods.

We observed in the experiments that the accuracy usually falls when there are limited AU occurring in the test data. This leads to high values in the estimated weights for training instances that were not reliable. We are currently working on this issue. Finally, it is worth pointing out that STM is a general framework with applicability beyond the AU detection problem, and could be easily applied to other domains such as object or activity recognition.

Acknowledgments

Research reported in this publication was supported in part by the National Institute of Mental Health of the National Institutes of Health under Award Number R01MH096951 and the National Science Foundation under the grant RI-1116583. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health or the National Science Foundation. We would also like to thank Tzu-Kuo Huang for helpful discussions.

References

[1] Y. Aytar and A. Zisserman. Tabula Rasa: Model transfer for object category detection. In *ICCV*, 2011. 2, 3

[2] M. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Automatic recognition of facial actions in spontaneous expressions. *Journal of Multimedia*, 1(6):22–35, 2006. 1, 2, 5, 7

[3] K. M. Borgwardt, A. Gretton, M. J. Rasch, H. P. Kriegel, B. Schölkopf, and A. J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):49–57, 2006. 2

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011. 4

[5] L. Bruzzone and M. Marconcini. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *PAMI*, 32(5):770–787, 2010. 3, 4, 6

[6] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. 6

[7] K.-Y. Chang, T.-L. Liu, and S.-H. Lai. Learning partially-observed hidden conditional random fields for facial expression recognition. In *CVPR*, 2009. 2

[8] O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5):1155–1178, 2007. 3, 5

[9] S. W. Chew, P. Lucey, S. Lucey, J. Saragih, J. F. Cohn, and S. Sridharan. Person-independent facial expression detection using constrained local models. In *AFGR*, 2011. 3

[10] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive SVMs. *JMLR*, 7:1687–1712, 2006. 6

[11] F. De la Torre and J. Cohn. Facial expression analysis. *Visual Analysis of Humans: Looking at People*, page 377, 2011. 2

[12] L. Duan, I. Tsang, and D. Xu. Domain transfer multiple kernel learning. *PAMI*, 34(3):465–479, 2012. 3

[13] M. Dudık, R. E. Schapire, and S. J. Phillips. Correcting sample selection bias in maximum entropy density estimation. In *NIPS*, 2005. 2

[14] P. Ekman and W. Friesen. Facial action coding system: A technique for the measurement of facial movement. *Consulting Psychologists Press*, 1978. 1

[15] J. Gorski, F. Pfeuffer, and K. Klamroth. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical Methods of Operations Research*, 66(3):373–407, 2007. 4

[16] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 2009. 3, 4, 5, 6

[17] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, 1999. 3, 4, 6

[18] A. Khosla, T. Zhou, T. Malisiewicz, A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012. 2, 3

[19] G. Littlewort, J. Whitehill, T. Wu, I. Fasel, M. Frank, J. Movellan, and M. Bartlett. The computer expression recognition toolbox (CERT). In *AFGR*, 2011. 7

[20] 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. 1, 5, 6

[21] A. Martinez and S. Du. A model of the perception of facial expressions of emotion by humans: Research overview and perspectives. *JMLR*, 13:1589–1608, 2012. 2

[22] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004. 2, 5

[23] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence. *Dataset shift in machine learning*. 2009. 3

[24] O. Rudovic, V. Pavlovic, and M. Pantic. Kernel conditional ordinal random fields for temporal segmentation of facial action units. In *ECCV Workshop*, 2012. 2

[25] J. Saragih, S. Lucey, and J. Cohn. Face alignment through subspace constrained mean-shifts. In *ICCV*, 2009. 2

[26] L. Shang and K. Chan. Nonparametric discriminant HMM and application to facial expression recognition. In *CVPR*, 2009. 2

[27] T. Simon, M. H. Nguyen, F. De la Torre, and J. F. Cohn. Au detection with segment-based SVMs. In *CVPR*, 2010. 2, 3

[28] M. Sugiyama, M. Krauledat, and K. Müller. Covariate shift adaptation by importance weighted cross validation. *JMLR*, 8:985–1005, 2007. 2

[29] M. Sugiyama, S. Nakajima, H. Kashima, P. Von Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NIPS*, 2008. 3

[30] Y. Tong, W. Liao, and Q. Ji. Facial action unit recognition by exploiting their dynamic and semantic relationships. *PAMI*, 29(10):1683–1699, 2007. 2

[31] A. Torralba and A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 2

[32] M. F. Valstar, M. Mehu, B. Jiang, M. Pantic, and K. Scherer. Meta-analysis of the first facial expression recognition challenge. *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, 42(4):966–979, 2012. 1, 2, 5, 6, 7

[33] M. F. Valstar and M. Pantic. Fully automatic recognition of the temporal phases of facial actions. *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, 42(1):28–43, 2012. 2

[34] T. Wu, N. J. Butko, P. Ruvolo, J. Whitehill, M. S. Bartlett, and J. R. Movellan. Action unit recognition transfer across datasets. In *Automatic Face & Gesture Recognition*, 2011. 3

[35] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, 2013. 2

[36] M. Yamada, L. Sigal, and M. Raptis. No bias left behind: Covariate shift adaptation for discriminative 3d pose estimation. In *ECCV*, 2012. 3

[37] P. Yang, Q. Liu, and D. N. Metaxas. Exploring facial expressions with compositional features. In *CVPR*, 2010. 2

[38] H. Zhang, A. C. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006. 3

[39] Y. Zhu, F. De la Torre, J. F. Cohn, and Y. J. Zhang. Dynamic cascades with bidirectional bootstrapping for action unit detection in spontaneous facial behavior. *IEEE Trans. on Affective Computing*, 2(2):79–91, 2011. 2, 5