Confidence Preserving Machine for Facial Action Unit Detection

Jiabei Zeng[1] **Wen-Sheng Chu**[2] **Fernando De la Torre**[2] **Jeffrey F. Cohn**[2,3] **Zhang Xiong**[1]
[1]School of Computer Science and Engineering, Beihang University, Beijing, China 100191
[2]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA 15213
[3]Department of Psychology, University of Pittsburgh, Pittsburgh, PA, USA 15260

Abstract

Varied sources of error contribute to the challenge of facial action unit detection. Previous approaches address specific and known sources. However, many sources are unknown. To address the ubiquity of error, we propose a Confident Preserving Machine (CPM) that follows an easy-to-hard classification strategy. During training, CPM learns two confident classifiers. A confident positive classifier separates easily identified positive samples from all else; a confident negative classifier does same for negative samples. During testing, CPM then learns a person-specific classifier using "virtual labels" provided by confident classifiers. This step is achieved using a quasi-semi-supervised (QSS) approach. Hard samples are typically close to the decision boundary, and the QSS approach disambiguates them using spatio-temporal constraints. To evaluate CPM, we compared it with a baseline single-margin classifier and state-of-the-art semi-supervised learning, transfer learning, and boosting methods in three datasets of spontaneous facial behavior. With few exceptions, CPM outperformed baseline and state-of-the art methods.

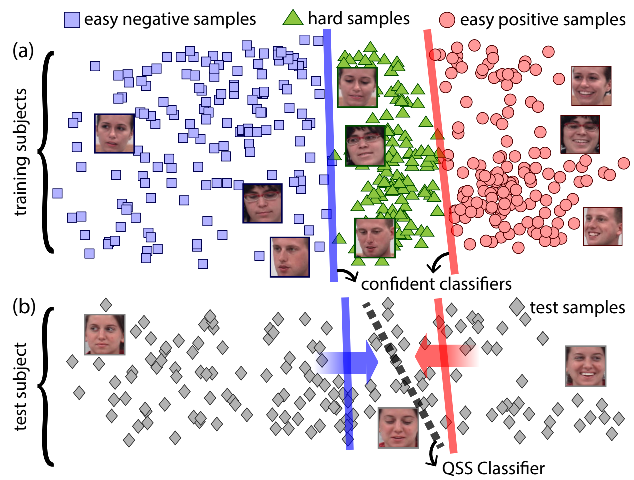

Figure 1. The main idea of the proposed Confidence Preserving Machine (CPM). (a) A pair of classifiers (red line and blue line), refered as *confident classifiers*, are learned to cooperatively separate easy and hard samples. (b) These confident classifiers are applied to recognize easy samples in a test subject. Then, a quasi-semi-supervised (QSS) classifier (black dash line) is learned by propagating predictions from the easy test samples to hard ones.

1. Introduction

Facial expressions convey varied and nuanced meanings. Small variations in timing and packaging of smiles, for instance, can communicate a polite greeting, felt enjoyment, embarrassment, or social discomfort. To analyze information afforded by facial expression, Ekman and Friesen proposed the Facial Action Coding System (FACS) [20]. FACS describes a facial activity in terms of anatomically based Action Units (AUs). AUs can occur alone or in combinations to represent nearly all possible facial expressions. AUs have a temporal envelope that minimally include an onset (or start) and an offset (or stop) and may include changes in intensity. There has been encouraging progress on facial AU detection during the past decades, especially for posed facial actions [11, 14, 17, 39, 44, 49].

Yet, accurate detection of spontaneous facial actions remains challenging [14, 32, 33, 41]. A number of sources of error have been identified. They include individual differences in participants (*e.g.*, gender, ethnicity), video resolution, head yaw, and low intensity. To model these variability, typically a highly non-linear decision boundary is necessary to infer a correct AU. A highly non-linear decision boundary typically lead to over-fitting, and it has been previously shown that existing algorithms generalize poorly to unseen subjects [10, 40]. Standard supervised approaches, such as SVM [21] and Boosting [23], use a single hyperplane to separate positive and negative samples. While these classifiers may perform well on samples with high-intensity AUs, frontal head pose or on particular subjects, they often fail with various appearance changes and subtle AUs (*i.e.*, low intensity). In this paper, we refer to these samples as *easy samples* and *hard samples*, respectively.

To reduce errors occasioned above, we propose a two-stage learning framework that combines multiple classifiers following an "easy-to-hard" strategy. This approach, which we refer to as a Confident Preserving Machine (CPM), is illustrated in Fig. 1. During training, CPM learns a pair of confident classifiers. One separates easy positive samples from all else. The other does the same for easy negative samples. During testing, CPM then learns a person-specific classifier using a quasi-semi-supervised (QSS) approach to propagate labels from easy samples to hard ones. Labels for QSS come from the confident classifiers and are referred to as virtual labels. In addition, we propose an iterative extension of CPM, termed as iCPM, which iterates between the confident classifiers learning in training and the QSS classifier in testing.

2. Related Work

Here we review related work in error reduction, semi-supervised learning, and transfer learning.

Error reduction: Previous efforts to reduce detection errors have focused on specific sources. To reduce error occasioned by subtle expressions, spatio-temporal directional features extracted by robust PCA [43] and temporal interpolation using {SVM,MKL,RF} classifiers [38] have been proposed. For error involving head pose, particle filters with multi-class dynamics [16] or variable-intensity templates [31] have been proposed. Individual differences in participants also have been considered. [10, 40] used a domain-transfer approach. In many cases, however, sources of error can be quite varied and even unknown as to their origin. CPM seeks to minimize error from all potential sources.

Semi-supervised learning (SSL): SSL has emerged an exciting field of incorporating unlabeled data for training. Such techniques make different assumptions on relationships between input and label space [7]. Smoothness assumption enforces data with same labels to be close to each other, and can be modeled by the prevalent graph-based method [34]. Cluster assumption employs the clustering behaviors of data with same labels. It has shown to be equivalent to low-density separation [8], and can be extended to entropy minimization [26]. Manifold assumption considers that high-dimensional data lie roughly on a low-dimensional manifold. Instead of Euclidean distances used in the smoothness assumption, manifold assumption considers metrics of manifold. Closest to our work is the Laplacian SVM (LapSVM) [3, 36], which incorporates the manifold assumption as a regularization for learning an SVM. Some other work explores the combination of the three assumptions in a boosting framework [9]. Interested readers are referred to [7, 48] for a more extensive review.

Notwithstanding the progress being made, these assumptions are unsuitable for AU detection, because subjects behave as different distributions in the feature space. Thus, closer data are unlikely to belong to the same label, *i.e.*, smoothness assumptions in SSL could be violated. On the contrary, CPM applies smoothness assumption to unlabeled test samples, where individual differences are excluded.

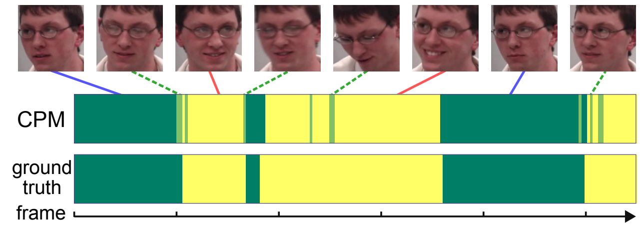

Figure 2. Illustration of using CPM on identifying AU12 from a real video. Dashed lines (light green) indicate the hard samples due to low intensities and head pose; solid lines indicate the easy samples for positive (light yellow) and negative (dark green) ones.

Transfer learning: Transfer learning also assumes different distributions between some training data and test data. The information between two different domains can be transferred by finding one or multiple intermediate spaces that minimize their 'mismatch'. Given each domain represented as a linear subspace, their similarities can be evaluated on aligned subspaces [22], or as their geodesic distances on a Grassmann manifold [24, 25]. The discrepancy between raw features can be alleviated by learning a transformation [29, 37]. Some seeks to the idea of *importance reweighting* to adapt one or multiple training domain(s) to a test domain [5, 28, 42]. Following this direction, Selective Transfer Machine (STM) [10] was proposed to remedy individual differences in facial AU detection by treating each subject as a domain. Recently, there have been several studies that describe a training domain as classifier parameters, and assume that an ideal classifier for the test domain can be represented as a combination of the learned classifiers [18, 19, 45]. Merging into this direction, STM was extended by transferring from source classifiers, and reduced training time complexity [40].

CPM differs from transfer learning in three ways. One, most transfer learning methods emphasize individual differences in subjects. CPM assumes that error has multiple sources. Individual differences are only one. Other sources include head pose and AU intensity. Two, CPM includes spatial-temporal smoothness that is absent in most transfer learning approaches. Three, CPM is more efficient, because it avoids the selection from multiple sources domains [19, 40] or re-weighting each sample [10, 42].

3. Confidence Preserving Machine (CPM)

3.1. Overview

Facial AU detection typically deals with data in the form of videos, *i.e.*, each subject has at least a clip of video instead of a single image. Among these videos, some frames are easier to tell an AU presence than others. Fig. 2 shows the easy and hard frames from a particular video. Because hard samples are intrinsically inseparable, treating easy and

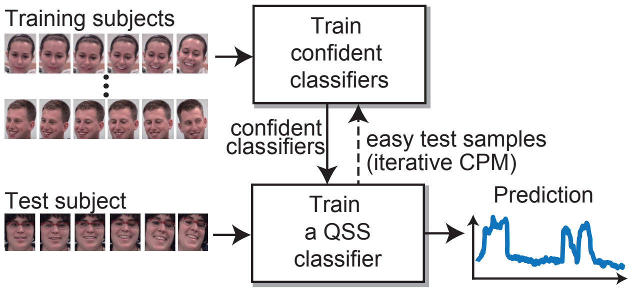

Figure 3. The proposed two-stage CPM framework: Given training videos, the confident classifiers are first trained, and then passed to train a QSS classifier, which makes the final prediction on a test subject. In iterative CPM, easy test samples are selected to iteratively augment the training set.

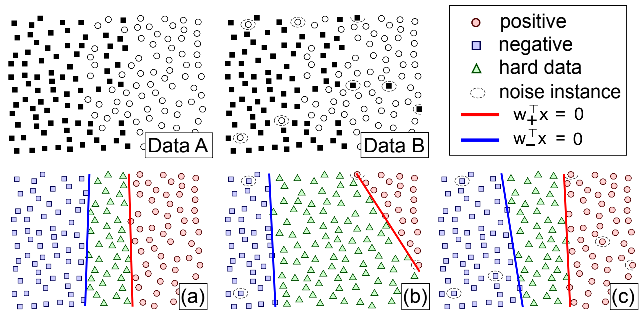

Figure 4. Illustration of two relabeling strategies. Data A and B are two synthetic data without and with noisy instances, respectively. (a)$\sim$(c) show the confident classifiers learned on the relabeled data using holistic relabeling on A, holistic relabeling on B, and localized relabeling on B, respectively.

hard samples equally would degrade the performance of a standard single-hyperplane classifier (*e.g.*, SVM [21]).

To address these issues, we propose the CPM, a two-stage framework that exploits multiple classifiers with an *easy-to-hard* strategy. Fig. 3 illustrates the CPM framework. The first stage, *training confident classifiers*, aims to find a pair of classifiers that distinguish easy and hard samples in training subjects. We define the easy samples as the ones on which the predictions of the confident classifiers agree with each other, and the hard samples otherwise. Compared to standard approaches that use a single classifier, each of confident classifiers focuses on predicting one class. The confident classifiers, therefore, are able to identify whether an unseen sample is easy or not, and predict confidently on it. In the second stage, *training a QSS classifier*, we first identify easy test samples by applying the trained confident classifiers. With confident predictions on easy test samples, a quasi semi-supervised (QSS) approach is introduced to train a person-specific classifier. The QSS classifier determines the label of the hard samples by propagating consistently the predictions in space and time.

3.2. Train confident classifiers

The first stage in CPM is to train the *confident classifiers*, a pair of classifiers that aims to cooperatively identify and separate easy and hard samples in the training set $\{\mathbf{x}_i, y_i\}_{i=1}^n$ with index $\mathcal{D} = \{1, 2, \ldots, n\}$, where $y_i \in \{+1, -1\}$ denotes a label and n is the size of training set.

In this paper, we cast the AU detection problem as a binary classification problem, although multi-label formulations have been proposed (*e.g.*, [47]). We formulate CPM in the context of maximum margin learning extending the support vector machine (SVM), but it can be applicable to any other supervised learning paradigm. The intuition behind the confident classifiers is to learn two classifiers, one for the positive class, represented by a hyperplane $\mathbf{w}_+$, and will predict confidently positive samples; similarly $\mathbf{w}_-$ is for the negative class.We will consider easy samples $\mathcal{E}$ as the subset of the training samples where both classifiers make the same prediction and *hard* samples $\mathcal{H}$ otherwise. It is important to note that $\mathbf{w}_+$ and $\mathbf{w}_-$ will classify the easy positive and negative samples respectively and they do not necessarily need to be parallel. Mathematically speaking,

$$\begin{cases} \mathcal{E} = \{i \in \mathcal{D} | y_i \mathbf{w}_y^\top \mathbf{x}_i > 0, \forall y \in \{+, -\}\}, \\ \mathcal{H} = \mathcal{D} \setminus \mathcal{E}, \end{cases} \tag{1}$$

where $\mathcal{E}$ and $\mathcal{H}$ denote the index sets of easy samples and hard samples, and we denote the confident classifiers $(\mathbf{w}_+, \mathbf{w}_-)$, or $\mathbf{w}_y$.

Learning the confident classifiers can be done iteratively by maximizing the margin as:

$$\begin{aligned} \min_{\mathbf{w}_y, \mathcal{E}} \quad & ||\mathbf{w}_y||^2 + \sum_{i,j} \left(\xi_i^2 + \xi_j^2\right) \qquad (2) \\ \text{s.t.} \quad & y_i \mathbf{w}_y^\top \mathbf{x}_i \geq 1 - \xi_i, \forall i \in \mathcal{E}, \\ & \eta_j^y \mathbf{w}_y^\top \mathbf{x}_j \geq 1 - \xi_j, \forall j \in \mathcal{H}, \end{aligned}$$

where y_i is the ground truth label, η_j^y is a *relabel* of a hard training sample $\mathbf{x}_j$ (explained below). ξ_i and ξ_j are non-negative slack variables for easy samples and hard samples respectively, to take into account misclassification. The easy samples, will preserve the original labels $\mathbf{y}_i$, whereas we will relabel the hard samples as η_j^+ for $\mathbf{w}_+$ and η_j^- for $\mathbf{w}_-$, to make the classifiers as confident as possible.

We present Alg. 1 to solve (2). Because the partition of hard samples $\mathcal{H}$ and easy samples $\mathcal{E}$ should be learned at the same time as confident classifiers, Alg. 1 updates $\mathcal{H}$, $\mathcal{E}$ and the confident classifiers $(\mathbf{w}_+, \mathbf{w}_-)$ alternatively. Note that we cannot guarantee a convergence of this process, thus a maximum iteration is set. The set of hard samples is initialized as empty. In the later iterations, hard samples are updated as those misclassified by both $\mathbf{w}_+$ and $\mathbf{w}_-$. The relabeling strategy enables $\mathbf{w}_+$ and $\mathbf{w}_-$ to preserve confident prediction in each class by adjusting the labels for hard samples. Here, we explore two relabeling strategies:

Algorithm 1 Train confident classifiers

Input: Data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ and its index set $\mathcal{D} = \{1, 2, \ldots, n\}$

Output: Confident classifiers $(\mathbf{w}_+, \mathbf{w}_-)$, easy samples $\mathcal{E}$ and hard samples $\mathcal{H}$

1: Initialization: $\mathcal{E} \leftarrow \mathcal{D}$; $\mathcal{H} \leftarrow \emptyset$;
2: **repeat**
3: $(\mathbf{w}_+, \mathbf{w}_-) \leftarrow$ solve (2) with fixed $\mathcal{E}$ and $\mathcal{H}$;
4: Update easy and hard samples $(\mathcal{E}, \mathcal{H})$ using (1);
5: Update relabels $\eta_j^+, \eta_j^- \forall j \in \mathcal{H}$;
6: **until** convergence or reach maximum #iteration

1) **Holistic relabeling**: The most straightforward strategy is to relabel *all* hard samples as $+1$ when training $\mathbf{w}_-$, and -1 when training $\mathbf{w}_+$, *i.e.*, $\eta_j^y = -y, \forall \mathbf{x}_j \in \mathcal{H}$. We term this strategy *holistic relabeling*. The main advantage of holistic relabeling is its low computational complexity.

2) **Localized relabeling**: Holistic relabeling may result in some unnecessary hard samples if signal noise exists. To gain more robustness to signal noise, we relabel an hard sample $\mathbf{x}_j$ as $+1$ *only when* there exists a neighboring support instance $\mathbf{x}_k$ with positive ground truth label, and similarly for relabeling $\mathbf{x}_j$ as -1. We term this *localized relabeling*. Denote the set of instances with support instances as $\mathcal{S}_y = \{j \in \mathcal{H} | \exists k \in \mathcal{H} : d(\mathbf{x}_j, \mathbf{x}_k) \leq r, y_k = y\}$, where r is a threshold and $d(\mathbf{x}_j, \mathbf{x}_k)$ is the distance between $\mathbf{x}_j$ and $\mathbf{x}_k$. The relabeling is formulated as

$$\eta_j^+ = \begin{cases} -1 & \mathbf{x}_j \in \mathcal{S}_- \\ y_j & \text{otherwise} \end{cases}, \quad \eta_j^- = \begin{cases} +1 & \mathbf{x}_j \in \mathcal{S}_+ \\ y_j & \text{otherwise} \end{cases}. \tag{3}$$

For simplicity, both strategies use binary labels. Note that other relabeling strategies are directly applicable, *e.g.*, weighting the relabels similar to those in DA-SVM [5], or introducing the concepts of bags as in MIL [1]. Fig. 4 illustrates the two relabeling strategies on synthetic examples. (a) and (b) illustrate the confident classifiers learned using holistic relabeling on A and B, respectively. As can be seen, the confident classifiers move toward the noisy instances in (b), showing that the holistic relabeling is improper for the presence of noise. Fig. 4(c) illustrates the result using localized relabeling, which is more robust to noisy instances.

3.3. Train a quasi-semi-supervised (QSS) classifier

In the previous section, we have discussed how to train the confident classifiers. As pointed out first by Chu *et al.* [10], a generic classifier trained on many subjects is unlikely to generalize well to an unseen subject because the training and test distributions could vary due to camera model, intra-personal variability, illumination, etc. Chu *et al.* [10] showed that person-specific and personalized models outperformed existing methods. Following this motivation, in this section, we train a quasi-semi-supervised (QSS) classifier with *virtual labels* provided by the confident classifiers. We term it QSS instead of semi-supervised because the labels are not provided in ground truth.

Recall our goal is to train a person-specific classifier $f_t(\mathbf{x}) = \mathbf{w}_t^\top \mathbf{x}$ for the test subject. To obtain such classifier, labels for the test subject are required. CPM collects such labels from the prediction of confident classifiers $\mathbf{w}_+$ and $\mathbf{w}_-$. Because confident classifiers are trained with many subjects, they are likely to generalize well to easy samples. However, on the other hand, there remains hard samples that CPM find difficult to identify. To disambiguate the hard samples, CPM adopts a quasi-semi-supervised (QSS) classifier that uses Laplacian similarity to enforce label smoothness on spatially and temporally neighboring samples.

Suppose we are given a test video with m frames denoted by $\mathbf{X}^{te} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m]^\top$ with index $\mathcal{D}^{te} = \{1, 2, \ldots, m\}$. CPM will first identify the easy test samples $\mathcal{E}_t$ as the ones on which both $\mathbf{w}_+$ and $\mathbf{w}_-$ agrees in the label prediction, *i.e.*, $\mathcal{E}_t = \{i \in \mathcal{D}^{te} | \operatorname{sign}(\mathbf{w}_+^\top \mathbf{x}_i) = \operatorname{sign}(\mathbf{w}_-^\top \mathbf{x}_i)\}$, and $\hat{y}_i = \operatorname{sign}(\mathbf{w}_y^\top \mathbf{x}_i)$ is a virtual label for an easy test sample. Once these virtual labels are obtained, CPM will propagate labels to the hard samples with a semi-supervised strategy minimizing:

$$\min_{\mathbf{w}^t} \sum_{i \in \mathcal{E}_t} \ell(\hat{y}_i, \mathbf{w}_t^\top \mathbf{x}_i) + \gamma_s ||\mathbf{w}_t||^2 + \gamma_I S(\mathbf{w}_t, \mathbf{X}^{te}), \tag{4}$$

where γ_s, γ_I control the importance of regularizations. $S(\mathbf{w}_t, \mathbf{X}^{te})$ is defined as the smoothness term that enforces the neighboring instances in both the feature space and the temporal space to have similar predictions:

$$S(\mathbf{w}_t, \mathbf{X}^{te}) = (\mathbf{X}^{te}\mathbf{w}_t)^\top \mathbf{D}^\top \mathbf{D} \mathbf{X}^{te} \mathbf{w}_t, \tag{5}$$

where $\mathbf{D}$ is a smoothness matrix that penalizes differences in the predictions of temporally and spatially adjacent instances. Specifically, $\mathbf{D}_{ii} = 1$, $\mathbf{D}_{ij} = -\frac{1}{Z_i} \lambda_{ij} e_{ij}$, $|i - j| \leq T, i \neq j$; λ is a Gaussian-like weight, such that closer frames have more similar predicted labels (see Fig. 5(a) for an illustration with $T = 5$). e_{ij} is 1 if $\|\mathbf{x}_i - \mathbf{x}_j\|_2 \leq \epsilon$, and 0 otherwise. It excludes the smoothness between the frames that are far away in feature space. Z_i is a normalization factor such that $\frac{1}{Z_i} \sum_{j=i-T, j \neq i}^{i+T} \lambda_{ij} e_{ij} = 1$. $\mathbf{D}_{ij} = 0$ elsewhere. We provide more derivation details in the supplementary material. Note that $\mathbf{D}^\top \mathbf{D}$ assembles Laplacian matrix by imposing smoothness on neighboring samples and are both positive semi-definite. However, $\mathbf{D}^\top \mathbf{D}$ considers both temporal and spatial constrains with Gaussian-like weight λ_{ij} and ejected factor e_{ij}, respectively.

Fig. 5 shows the effectiveness of the smoothness term S on 3 AUs in the BP4D dataset. To start the label propagation, 2.5% frames were randomly selected from each video as the estimated labels of easy instances. We compare the prediction on the rest 97.5% frames by training a

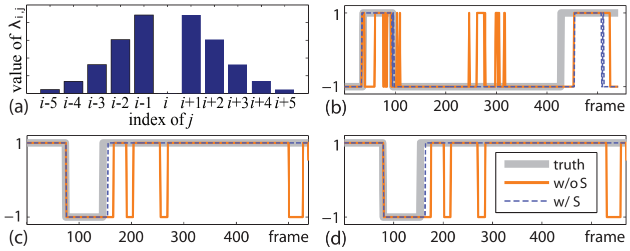

Figure 5. (a) an example of λ with $T=5$. (b)$\sim$(d) shows the effectiveness of smoothness term S on AU6 on video `2F01_11`, AU12 on video `2F01_09`, and AU17 on video `2F01_09`, respectively. The y-axis denotes AU occurrence (+1: presence, -1: absence).

linear SVM only using the labeled frames, and one with the smoothness term S over all the labeled and unlabeled data. As can be seen, compared to the ground truth, the prediction with the smoothness term performs more consistent result across 3 AUs. Although being rare, in some cases, it it possible that easy test samples are unavailable. Consequently, Eq. (4) fails to learn a QSS classifier $\mathbf{w}_t$. In this case, we simply assign $\mathbf{w}_t = \frac{1}{2}(\mathbf{w}_+ + \mathbf{w}_-)$.

3.4. Iterative CPM

CPM learns in sequential fashion the confident classifiers (Sec. 3.2) and the QSS classifier (Sec. 3.3). So, the PS classifier learned in a QSS fashion depends indirectly on the training data through the confident classifiers. However, it is likely that there is mismatch between the training and testing [10, 40], and the confident classifiers might not generalize well even in the easy samples. To address this issue, we propose iterative CPM (iCPM) that jointly learns the confident and PS classifiers.

In the iCPM, at each iteration, the easy test samples are selected to be part of the training for the confident classifiers, so the confident classifiers are trained with test data (but no labels of test data are provided). Alg. 2 summarizes the steps for the iCPM algorithm. Fig. 6 illustrates a synthetic example. In Fig. 6, the training and test distribution are different. In the initialized iteration (0), all training data is labeled as easy samples, so the confident classifiers are basically a standard SVM, and $\mathbf{w}_+$ and $\mathbf{w}_-$ are the same. This classifier achieves 97% accuracy on test data. In the first iteration (1), we update the hard-samples (green triangles) and re-train the confident classifiers. The CPC and CNC identify easy samples (blue and red diamonds) in test data, and the QSS classifier labels the hard samples (green diamonds), and learns the hyperplane (black line). At the second iteration(2), the classifier achieves 99% of accuracy. Finally, at the third iteration, the easy and hard samples are again updated to train $(\mathbf{w}_+, \mathbf{w}_-)$ and QSS classifier achieving 100% of classification accuracy.

Complexity: As in standard transfer learning methods [18, 42], iCPM incorporates all the training data to

Algorithm 2 Iterative Confidence Preserving Machine

Input: labeled training data $\{\mathbf{x}_i, y_i\}_{i=1}^n$ with index set $\mathcal{D} = \{1, 2, \ldots, n\}$, unlabeled test data $\{\mathbf{x}_i^{te}\}_{i=1}^m$ with index set $\mathcal{D}^{te} = \{1, 2, \ldots, m\}$

Output: QSS classifier $\mathbf{w}_t$

$\mathcal{E} \leftarrow \mathcal{D}, \mathcal{H} \leftarrow \emptyset$;
$(\mathbf{w}_+, \mathbf{w}_-) \leftarrow$ solve (2);
$(\mathcal{E}, \mathcal{H})$ using (1);
repeat
  Update relabels $\eta_j^+, \eta_j^-, \forall j \in \mathcal{H}$;
  $(\mathbf{w}_+, \mathbf{w}_-) \leftarrow$ solve (2) with fixed $\mathcal{E}$ and $\mathcal{H}$;
  Estimate virtual labels $\{\hat{y}_i\}_{i=1}^m$,
$$\hat{y}_i = \begin{cases} 1 & \mathbf{w}_y^\top \mathbf{x}_i^{te} > 0, \forall y \in \{-1, +1\}, \\ -1 & \mathbf{w}_y^\top \mathbf{x}_i^{te} < 0, \forall y \in \{-1, +1\}, \\ 0 & \text{otherwise.} \end{cases}$$
  $\mathcal{E}_t = \{i \in \mathcal{D}^{te} | \operatorname{sign}(\mathbf{w}_+^\top \mathbf{x}_i^{te}) = \operatorname{sign}(\mathbf{w}_-^\top \mathbf{x}_i^{te})\}$;
  if $\exists i, j \in \mathcal{E}_t$, s.t. $\hat{y}_i = -1, \hat{y}_j = 1$ **then**
    $\mathbf{w}_t \leftarrow$ solve (4) given $\mathbf{X}^{te}$ and $\{\hat{y}_i\}_{i=1}^m$;
  else
    $\mathbf{w}_t = \frac{1}{2}(\mathbf{w}_+ + \mathbf{w}_-)$;
  end if
  Update $\mathcal{E}_t = \{i \in \mathcal{D}^{te} | \hat{y}_i = \operatorname{sign}(\mathbf{w}_t^\top \mathbf{x}_i^{te})\}$;
  Update $(\mathcal{E}, \mathcal{H}) \leftarrow (1)$;
  $\mathcal{E} \leftarrow \mathcal{E} \cup \mathcal{E}_t$;
until convergence

compute a QSS classifier for each test clip. Despite so, iCPM is relatively efficient in training due to the learning of linear classifiers . In Alg. 2, solving (2) with fixed $\mathcal{E}$ and $\mathcal{H}$ and solving (4) are both linear with complexity $\mathcal{O}(\max(n, d) \min(n, d)^2)$ [6], where d is the dimension of features; n is the number of samples in $\mathcal{E} \cup \mathcal{H}$ in (2), or the number of test samples in (4).

3.5. Comparison with alternative methods

Besides CPM and iCPM, concepts similar to easy and hard samples have presented in other methods. Boosting methods learn a strong classifier after combining a set of weak classifiers. However, it fits a classifier for completely labeled data without coping with unlabeled data. CPM or iCPM also seems to be like co-training [4], which alternatively trains two or more classifiers so that the most confident samples from one classifier are used to train another. But in co-training, labeled data and unlabeled data are supposed to have a same distribution.

As a component of CPM, confident classifiers are similar to SVM with reject options (RO-SVM) [2, 27], which designs new loss functions where data in reject region have a loss value between 0 and 1. We can think of RO-SVM as learning two parallel hyperplanes between which lie the hard samples. Unlike RO-SVM, hyperplanes in the pro-

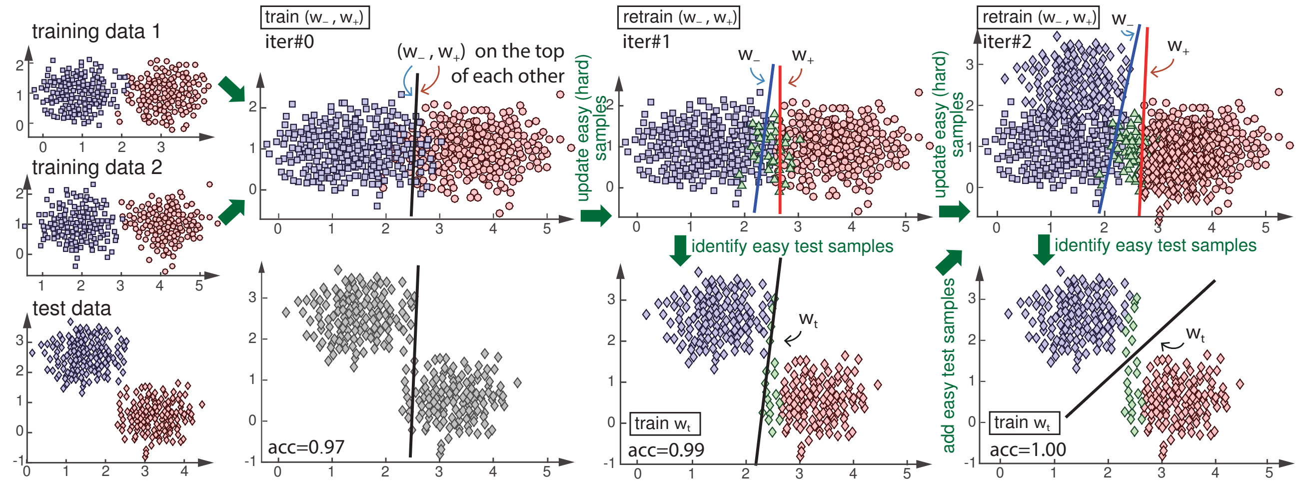

Figure 6. A toy example of iCPM. The first column illustrates two training subjects (rectangles and circles) and a test subject (diamonds). A same color means a same class. The second, third, and forth column illustrates the initialization and two iterations in Alg. 2, respectively. Points in blue and red colors are easy data, while those in green are hard ones.

posed confident classifiers are not necessarily paralleled. Twin SVM (TW-SVM) [30] also has two hyperplanes, where each plane is close to one class and far from the other. Confident classifiers are different from TW-SVM due to their different purposes. TW-SVM aims to make more accurate predictions on all the samples but cannot tell hard samples from easy ones, while confident classifiers are obligated to distinguish hard and easy samples, and only predict on easy ones.

4. Experiments

4.1. Datasets

GFT [12] are recorded when three previously unacquainted young adults sat around a circular table for 30-min conversation with drinks. Moderate out-of-plane head motion and occlusion are presented in the videos which makes the AU detection challenging. In our experiments, 50 subjects are selected and each video is about 5000 frames.

BP4D [46] is a spontaneous facial expression dataset in both 2D and 3D videos. The dataset includes 41 participants aging from 18 to 29 associating with 8 tasks, which are covered with an interview process and a series of activities to elicit eight emotions. Frame-level ground-truth for facial actions are obtained using the Facial Action Coding System. In our experiments, we only use the 2D videos.

DISFA [35] recorded 27 subjects' spontaneous expressions when they were watching video clips. DISFA not only codes the AUs, but also labels the intensities. In our experiments, we use the frames with intensities equal or greater than A-level as positive, the rest as negative. The dataset consist of 27 videos with 4845 frames each.

4.2. Settings

All the experiments were conducted using same protocol for fairness. Each dataset was divided into 10 splits, where each split designated several (5 in GFT, 4 or 5 in BP4D, 2 or 3 in DISFA) subjects as test data and the remaining as training data. Each subject served as test data once during the ten splits. 49 landmarks in the face were tracked by IntraFace [13]. For each AU, SIFT descriptors around the associated landmarks were extracted, *e.g.*, the landmarks around the mouth for AU12. The same feature were used throughout the experiments.

We evaluated the performance using frame-based F1-score (F1-frame), which is prevalent in binary classification problems, and event-based F1 (F1-event) [15], which evaluates detection at event-level. An event is defined as a max continuous period that an AU is present. F1-Event is similar to F1-frame but applying event-based precision EP and recall ER, *i.e.*, F1-event= $\frac{2EP \cdot ER}{EP+ER}$. An event-level agreement holds if the overlap of two events is above a certain threshold. Both F1-frame and F1-event were reported for each AU and averaged over all the AUs.

4.3. Objective evaluation on CPM components

Recall that two major components in CPM are the confident classifiers and the PS classifier learned with QSSL. In order to validate their effectiveness, we conducted comparisons with a baseline linear SVM [21], confident classifiers only (Conf), and CPM (Conf+PS classifiers). In Conf, we trained confident classifiers using Alg. 1, and then passed them to train a PS classifier without a smoothness assumption. Thus, Conf checks the effectiveness of confident classifiers when compared with a standard single-hyperplane SVM. CPM differs from Conf by learning the PS classifier with the spatial-temporal smoothness as discussed in

Table 1. Comparison on GFT. ("H" stands for an extra post-processing with HMM)

AU	F1-frame							F1-event						
	SVM|H	Ada|H	Lap|H	DAM	MDA	GFK	iCPM	SVM|H	Ada|H	Lap|H	DAM	MDA	GFK	iCPM
1	30.3|16.8	20.3|15.4	12.1|16.4	1.7	29.2	**30.9**	29.9	20.3|17.9	15.3|**28.2**	5.4| 9.7	2.1	21.3	21.6	27.1
2	25.6|18.4	14.8|21.8	26.0|19.3	5.3	25.8	**29.3**	25.7	20.2|21.1	12.2|**30.7**	18.2|16.6	4.7	21.3	22.5	24.8
6	66.2|66.4	62.1|47.3	2.7|40.7	58.0	63.8	66.1	**67.3**	49.1|**56.8**	47.5|43.4	4.4|37.5	50.0	47.0	50.2	**56.8**
7	70.9|72.2	69.6|50.0	24.0|50.3	66.0	66.6	72.2	**72.5**	50.4|59.8	50.7|44.0	21.6|48.3	41.7	49.2	52.1	**60.1**
10	65.5|65.5	65.5|43.7	56.7|61.2	64.9	65.4	**67.5**	67.0	50.2|57.8	50.2|46.6	46.5|57.5	53.1	51.6	54.3	**58.1**
12	74.2|**75.9**	73.0|54.5	64.8|69.0	72.9	71.9	72.7	75.1	56.3|**65.0**	54.7|59.9	54.9|64.4	61.9	52.0	54.3	**65.0**
14	79.6|78.1	77.7|59.2	76.7|51.2	79.5	74.0	79.8	**80.7**	63.8|70.8	62.3|59.9	**81.5**|61.2	64.6	63.7	64.8	74.7
15	34.1|17.5	20.3|20.5	19.3|13.9	1.4	31.8	31.7	**43.5**	28.1|20.1	17.7|41.8	15.9|20.2	2.3	25.4	26.8	**32.2**
17	49.2|**50.6**	48.2|38.6	42.5|21.2	34.6	47.4	48.9	49.1	42.9|**53.1**	37.1|38.5	36.4|25.9	29.6	41.4	41.3	52.3
23	28.3|29.8	19.4|20.7	27.1|25.1	2.8	26.0	26.7	**35.0**	27.7|35.9	16.8|**36.7**	9.5|19.5	4.4	26.7	27.1	25.9
24	31.9|21.0	22.3|25.8	25.7|16.9	3.0	31.8	**33.0**	31.6	30.3|21.8	20.8|26.4	21.7|13.9	4.9	30.0	30.5	**31.8**
Av.	48.7|46.6	44.8|36.1	32.8|35.0	35.5	48.5	48.6	**52.5**	38.6|43.7	35.0|41.5	27.3|34.1	29.0	39.1	38.9	**46.3**

Table 2. Comparison on BP4D. ("H" stands for an extra post-processing with HMM)

AU	F1-frame							F1-event						
	SVM|H	Ada|H	Lap|H	DAM	MDA	GFK	iCPM	SVM|H	Ada|H	Lap|H	DAM	MDA	GFK	iCPM
1	46.0|43.4	41.5|37.7	43.8|29.0	38.2	39.6	42.4	**46.6**	29.2|38.1	29.8|**41.7**	29.2|27.8	26.7	30.5	29.7	35.3
2	38.5|38.4	12.4|25.5	17.6|27.8	27.3	37.0	35.8	**38.7**	29.3|**36.1**	12.9|32.4	24.8|27.1	12.3	28.2	28.9	32.5
4	**48.5**|41.6	39.4|30.4	27.2|26.1	29.1	45.7	47.3	46.5	33.5|37.4	28.9|28.3	30.5|26.5	22.3	32.8	32.8	**39.4**
6	67.0|62.0	**71.7**|61.2	71.5|26.1	67.5	69.2	71.2	68.4	53.7|37.4	54.4|58.5	53.7|26.5	55.4	52.9	54.4	**60.9**
7	72.2|56.5	**74.7**|53.7	71.6|52.2	72.6	70.2	72.5	73.8	59.0|55.3	55.2|49.2	56.2|57.6	61.1	58.4	54.9	**62.1**
10	72.7|54.6	**75.7**|62.1	72.8|55.3	74.4	71.0	74.2	74.1	61.3|52.8	59.3|67.8	60.7|60.6	**68.6**	57.5	59.7	65.1
12	83.6|65.4	84.3|62.6	84.3|55.3	76.4	81.8	83.9	**84.6**	62.5|52.8	63.9|60.8	64.2|60.6	60.8	59.9	65.6	**71.4**
14	59.9|49.2	61.0|50.9	**62.6**|26.3	59.9	57.8	57.2	62.2	49.5|46.3	51.7|**56.7**	51.9|26.9	53.3	50.2	48.7	55.9
15	41.1|39.9	30.6|30.4	35.2|25.5	15.9	41.4	40.6	**44.3**	33.7|**39.0**	24.4|**39.0**	25.4|25.4	12.7	28.2	31.1	37.4
17	55.6|57.8	56.6|47.8	**59.1**|46.3	52.9	50.1	55.4	57.5	46.0|**56.1**	44.0|51.5	44.0|41.7	51.5	39.6	44.0	49.9
23	40.8|39.4	33.0|32.8	33.6|27.6	3.9	36.2	39.9	**41.7**	36.4|**44.0**	28.2|41.4	27.2|22.2	5.8	30.7	33.3	41.9
24	**42.1**|19.3	34.2|26.7	40.5|16.9	4.9	41.1	41.7	39.7	37.7|16.0	30.9|35.7	34.8|13.8	3.6	35.4	35.6	**38.7**
Av.	55.7|47.3	51.3|43.5	54.7|36.9	42.6	53.4	55.2	**56.5**	44.3|44.8	40.3|46.9	41.9|36.5	36.2	42.0	43.2	**49.2**

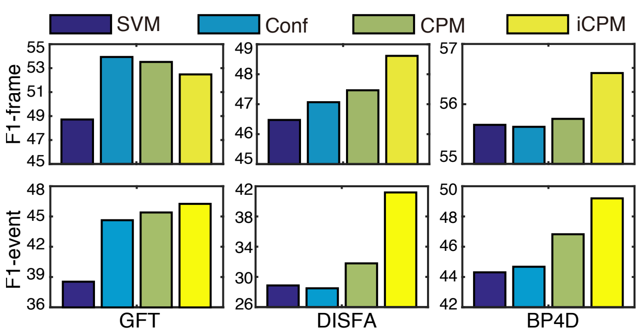

Figure 7. Results on GFT, DISFA and BP4D datasets. Note that the scales in each dataset are different for display purpose.

Sec. 3.3. In this way, CPM verifies the QSS classifier's effectiveness on propagating labels with smoothness assumptions. We also conducted iCPM to validate the iterative integration in CPM.

Fig. 7 illustrates the results on GFT, BP4D and DISFA datasets, respectively. The values of F1-frame and F1-event were reported as the average over all AUs. Comparing the results between SVM and Conf, confident classifiers showed positive affects on the performance. The effectiveness of applying smoothness assumptions was indicated by the results between Conf and CPM. Out of the results, iCPM outperformed CPM in most cases, validating the effectiveness of the proposed iterative integration.

4.4. Comparisons

This section compares the proposed CPM with alternative methods, including baseline single-hyperplane classifiers, semi-supervised learning (SSL), and transfer learning approaches. For baselines, we used LibLinear [21] and Matlab toolbox for Adaboost [23]. For SSL, we implemented a linear version of Laplacian SVM (Lap) [36]. Its kernel version is computationally prohibitive because our experiments contain more than 100,000 samples. For transfer learning, we compared to state-of-the-art methods including Geodesic Flow Kernel (GFK) [24], Domain Adap-

Table 3. Comparison on DISFA. ("H" stands for an extra post-processing with HMM)

<table>
<tr><th rowspan="2">AU</th><th colspan="7">F1-frame</th><th colspan="7">F1-event</th></tr>
<tr><th>SVM|H</th><th>Ada|H</th><th>Lap|H</th><th>DAM</th><th>MDA</th><th>GFK</th><th>iCPM</th><th>SVM|H</th><th>Ada|H</th><th>Lap|H</th><th>DAM</th><th>MDA</th><th>GFK</th><th>iCPM</th></tr>
<tr><td>1</td><td>26.5|14.4</td><td>17.1|12.4</td><td>13.1|16.2</td><td>7.9</td><td>19.0</td><td>23.2</td><td>29.5</td><td>14.5|17.6</td><td>16.1|21.2</td><td>9.6|11.6</td><td>5.4</td><td>11.6</td><td>18.1</td><td>18.7</td></tr>
<tr><td>2</td><td>24.0|15.3</td><td>20.1|10.5</td><td>6.4|12.6</td><td>13.1</td><td>9.5</td><td>16.3</td><td>24.8</td><td>10.8|17.6</td><td>17.3|17.0</td><td>11.4|11.0</td><td>12.1</td><td>16.4</td><td>17.4</td><td>19.2</td></tr>
<tr><td>4</td><td>56.1|48.5</td><td>59.8|26.4</td><td>21.1|23.4</td><td>40.4</td><td>59.3</td><td>60.3</td><td>56.8</td><td>31.6|37.2</td><td>32.5|27.7</td><td>15.9|16.2</td><td>32.4</td><td>28.6</td><td>28.3</td><td>41.8</td></tr>
<tr><td>6</td><td>40.9|34.9</td><td>31.9|22.1</td><td>22.1|19.9</td><td>19.2</td><td>21.1</td><td>41.9</td><td>41.7</td><td>30.3|29.2</td><td>28.3|25.9</td><td>23.7|13.5</td><td>22.6</td><td>30.8</td><td>30.6</td><td>36.9</td></tr>
<tr><td>9</td><td>30.5|10.9</td><td>29.3|17.4</td><td>12.1|10.9</td><td>11.9</td><td>7.6</td><td>30.3</td><td>31.5</td><td>23.4|13.2</td><td>22.7|39.9</td><td>7.8| 8.3</td><td>14.3</td><td>27.4</td><td>14.6</td><td>31.7</td></tr>
<tr><td>12</td><td>65.6|70.1</td><td>69.4|46.3</td><td>33.7|32.2</td><td>50.9</td><td>63.1</td><td>69.6</td><td>71.9</td><td>49.9|57.1</td><td>51.9|61.6</td><td>33.2|20.7</td><td>44.3</td><td>42.1</td><td>46.2</td><td>56.6</td></tr>
<tr><td>25</td><td>78.3|84.1</td><td>83.9|70.5</td><td>35.3|30.3</td><td>56.2</td><td>81.3</td><td>80.0</td><td>81.6</td><td>31.9|76.7</td><td>38.0|58.8</td><td>42.5|21.2</td><td>56.4</td><td>46.5</td><td>36.1</td><td>76.7</td></tr>
<tr><td>26</td><td>50.0|51.5</td><td>59.6|50.5</td><td>18.9|25.5</td><td>43.2</td><td>51.1</td><td>54.6</td><td>51.3</td><td>38.6|51.7</td><td>38.7|49.5</td><td>48.7|18.4</td><td>38.4</td><td>36.4</td><td>37.3</td><td>47.7</td></tr>
<tr><td>Av.</td><td>46.5|41.2</td><td>46.6|32.0</td><td>20.3|21.4</td><td>30.4</td><td>39.0</td><td>47.0</td><td>48.6</td><td>28.9|37.5</td><td>30.7|37.7</td><td>24.1|15.1</td><td>28.2</td><td>30.0</td><td>28.6</td><td>41.2</td></tr>
</table>

tion Machine (DAM) [18], and Multi-source Domain Adaptation (MDA) [42]. GFK computed the geodesic flow kernel from training to test sample, and then used it as a kernel in SVM. DAM fitted a classifier for test subject as a linear combination of classifiers of training subjects. Note that DAM is able to tackle with unlabeled test data. We did not use its extended version DSM [19] because DSM requires to enumerate all the possible selections of source domains, which are as much as 2^{45} in our experiment. MDA performed unsupervised domain adaptation by re-weighting both source domains and training instances. All methods, except for SVM and Ada, learned a specific classifier for each test subject. Codes of other competitive methods were either downloaded from author's webpage or provided by the authors. To show a more fair comparison, we also implemented Hidden Markov Model (HMM) as a post-processing for smoothing the prediction of SVM, Lap, and Ada. Note that HMM was not directly applicable for DAM, MDA, and GFK because their scores of the frame-level labeling output were available only for test data.

Tables 1∼3 show the results reported with the best parameters. SVM and Ada outperformed well in some AUs. Despite this, the overall performances of Ada were worse than iCPM, because Adaboost is a supervised method without investigating unlabeled test data. Overall, Lap had the worst performance due to its unsuitable assumption for spontaneous facial expression detection, which enforced the data to have similar decision values with their neighbors. Such assumption was not guaranteed across training and test subjects drawn from different distributions. Lap achieved better results on one or two AUs in BP4D. This is because most frames in BP4D dataset were frontal and thus had less appearance differences.

Both DAM and MDA assumed the QSS classifier is a linear combination of multiple source classifiers. When positive and negative data were extremely imbalanced, *e.g.*, AU1 on GFT, DAM performed poorly because each source classifier was unreliable. MDA performed better than DAM because MDA learned the weights for training data and source-domains instead of using fixed weights, meanwhile, MDA had a smooth assumption over test data. GFK performed similarly to SVM, although it did not provide a way to deal with multiple sources. Across three datasets, iCPM consistently outperformed three transfer learning methods.

With few exceptions, iCPM consistently outperformed the alternative methods in both metrics. Because iCPM incorporated the spatial-temporal smoothness term (as described in Sec. 3.3), it showed an obvious increase on F1-event compared to F1-frame. Recall that AU detection aims for detecting temporal events, we believe this spatial-temporal smoothness would significantly improve the detection result. Note that the experiments with HMM did not show consistent improvements on either F1-frame or F1-event as iCPM did. A possible explanation is that a trivial enforcement of temporal consistency is likely to make some frames similar to their misclassified neighbors, or over-smooth some short events. It indicated that the performance edge of iCPM was given by both separating easy/hard samples and its temporal-spatial smoothness.

5. Conclusion

We have presented the CPM for facial AU detection. Unlike standard methods with assumptions on sources of error, CPM censors hard-to-recognize samples that could be ascribed to low intensities, head motion, or individual differences. CPM exploits an easy-to-hard framework that incorporates the proposed confident classifiers and a quasi semi-supervised classifier regularized with spatial-temporal smoothness. We also introduce iCPM, an iterative extension of CPM, that gradually adds easy test samples to update the confident classifiers. Experiments on three spontaneous datasets showed the effectiveness of CPM against semi-supervised learning and transfer learning methods. Future work includes a non-linear extension of CPM.

Acknowledgements: This publication was supported in part by National Institutes of Health Award Number MH096951, National Natural Science Foundation of China (No. 61272350), National High Technology Research and Development Program of China (No. 2013AA01A603).

References

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, pages 561–568, 2002.

[2] P. L. Bartlett and M. H. Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9:1823–1840, 2008.

[3] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *CoLT*, pages 92–100, 1998.

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

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

[7] O. Chapelle, B. Schölkopf, A. Zien, et al. *Semi-supervised learning*, volume 2. MIT press Cambridge, 2006.

[8] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. *AISTATS*, 2005.

[9] K. Chen and S. Wang. Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions. *TPAMI*, 33(1):129–143, 2011.

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

[11] J. F. Cohn and F. De la Torre. *The Oxford Handbook of Affective Computing*, chapter Automated Face Analysis for Affective Computing. 2014.

[12] J. F. Cohn and M. A. Sayette. Spontaneous facial expression in a small group can be automatically measured: An initial demonstration. *Behavior Research Methods*, 42(4):1079–1086, 2010.

[13] F. De la Torre, W.-S. Chu, X. Xiong, F. Vicente, X. Ding, and J. F. Cohn. Intraface. In *Automatic Face and Gesture Recognition*, 2015.

[14] F. De la Torre and J. F. Cohn. *Guide to Visual Analysis of Humans: Looking at People*, chapter Facial Expression Analysis. Springer, 2011.

[15] X. Ding, W.-S. Chu, F. De la Torre, J. F. Cohn, and Q. Wang. Facial action unit event detection by cascade of tasks. In *ICCV*, 2013.

[16] F. Dornaika and F. Davoine. Simultaneous facial action tracking and expression recognition in the presence of head motion. *IJCV*, 76(3):257–281, 2008.

[17] S. Du, Y. Tao, and A. M. Martinez. Compound facial expressions of emotion. *PNAS*, 111(15):E1454–E1462, 2014.

[18] L. Duan, I. W. Tsang, D. Xu, and T.-S. Chua. Domain adaptation from multiple sources via auxiliary classifiers. In *ICML*, 2009.

[19] L. Duan, D. Xu, and S.-F. Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *CVPR*, 2012.

[20] P. Ekman and E. L. Rosenberg. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, 1997.

[21] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[22] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2013.

[23] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[24] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.

[25] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, 2011.

[26] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *NIPS*, 2005.

[27] Y. Grandvalet, A. Rakotomamonjy, J. Keshet, and S. Canu. Support vector machines with a reject option. In *NIPS*, pages 537–544, 2009.

[28] 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.

[29] I.-H. Jhuo, D. Liu, D. Lee, and S.-F. Chang. Robust visual domain adaptation with low-rank reconstruction. In *CVPR*, 2012.

[30] R. Khemchandani, S. Chandra, et al. Twin support vector machines for pattern classification. *TPAMI*, 29(5):905–910, 2007.

[31] S. Kumano, K. Otsuka, J. Yamato, E. Maeda, and Y. Sato. Pose-invariant facial expression recognition using variable-intensity templates. *IJCV*, 83(2):178–194, 2009.

[32] Y. Li, J. Chen, Y. Zhao, and Q. Ji. Data-free prior model for facial action unit recognition. *IEEE Trans. on Affective Computing*, 4(2):127–141, 2013.

[33] M. Liu, S. Shan, R. Wang, and X. Chen. Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition. In *CVPR*, 2014.

[34] W. Liu, J. Wang, and S.-F. Chang. Robust and scalable graph-based semi-supervised learning. *Proceedings of the IEEE*, 100(9):2624–2638, 2012.

[35] S. Mavadati, M. Mahoor, K. Bartlett, P. Trinh, and J. Cohn. Disfa: A spontaneous facial action intensity database. *IEEE Trans. on Affective Computing*, 4(2):151–160, April 2013.

[36] S. Melacci and M. Belkin. Laplacian support vector machines trained in the primal. *Journal of Machine Learning Research*, 12:1149–1184, 2011.

[37] K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *ICML*, 2013.

[38] T. Pfister, X. Li, G. Zhao, and M. Pietikainen. Recognising spontaneous facial micro-expressions. In *ICCV*, 2011.

[39] O. Rudovic, V. Pavlovic, and M. Pantic. Context-sensitive dynamic ordinal regression for intensity estimation of facial action units. *TPAMI*, 2015.

[40] E. Sangineto, G. Zen, E. Ricci, and N. Sebe. We are not all equal: Personalizing models for facial expression analysis with transductive parameter transfer. 2014.

[41] L. Shang and K.-P. Chan. Nonparametric discriminant hmm and application to facial expression recognition. In *CVPR*, 2009.

[42] Q. Sun, R. Chattopadhyay, S. Panchanathan, and J. Ye. A two-stage weighting framework for multi-source domain adaptation. In *NIPS*, 2011.

[43] S.-J. Wang, W.-J. Yan, G. Zhao, X. Fu, and C.-G. Zhou. Micro-expression recognition using robust principal component analysis and local spatiotemporal directional features. In *ECCV Workshops*. 2014.

[44] J. Whitehill, M. S. Bartlett, and J. R. Movellan. *Social Emotions in Nature and Artifact*, chapter Automatic facial expression recognition. 2014.

[45] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive SVMs. In *ACM MM*, 2007.

[46] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, and P. Liu. A high-resolution spontaneous 3d dynamic facial expression database. In *Automatic Face and Gesture Recognition*, 2013.

[47] K. Zhao, W.-S. Chu, F. De la Torre Frade, J. Cohn, and H. Zhang. Joint patch and multi-label learning for facial action unit detection. In *CVPR*, 2015.

[48] X. Zhu. Semi-supervised learning. In C. Sammut and G. Webb, editors, *Encyclopedia of Machine Learning*, pages 892–897. 2010.

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