Joint Patch and Multi-label Learning for Facial Action Unit Detection

Kaili Zhao[1] Wen-Sheng Chu[2] Fernando De la Torre[2] Jeffrey F. Cohn[2,3] Honggang Zhang[1]
[1]School of Comm. and Info. Engineering, Beijing University of Posts and Telecom., Beijing China
[2]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213
[3]Department of Psychology, University of Pittsburgh, Pittsburgh, PA 15260

Abstract

The face is one of the most powerful channel of non-verbal communication. The most commonly used taxonomy to describe facial behaviour is the Facial Action Coding System (FACS). FACS segments the visible effects of facial muscle activation into 30+ action units (AUs). AUs, which may occur alone and in thousands of combinations, can describe nearly all-possible facial expressions. Most existing methods for automatic AU detection treat the problem using one-vs-all classifiers and fail to exploit dependencies among AU and facial features. We introduce joint-patch and multi-label learning (JPML) to address these issues. JPML leverages group sparsity by selecting a sparse subset of facial patches while learning a multi-label classifier. In four of five comparisons on three diverse datasets, CK+, GFT, and BP4D, JPML produced the highest average F1 scores in comparison with state-of-the art.

1. Introduction

The Facial Action Coding System (FACS) [10] is a comprehensive system for describing facial movements. Anatomically-based descriptors, referred to as Action Units (AUs), alone and in thousands of combinations can account for nearly all-possible facial expressions. This descriptive power is not without cost. Manual FACS coding is labor intensive. Training can require a hundred hours or more to reach acceptable competence. Once a FACS coder achieves this milestone, annotation (also referred to as coding) can require an hour or more for each 30- to 60 seconds of video, and inter-observer reliability must be closely monitored to maintain quality. To make possible more efficient use of FACS, computer vision strives for automatic AU coding. While significant progress has been made toward this goal [1,6,9,22], at least two critical problems remain. These are patch and multi-label learning. Patch learning *(PL)* addresses how to effectively exploit local dependencies between features; multi-label learning *(ML)* seeks to exploit strong correlations among AUs.

Most current approaches extract features across the entire face and concatenate them for AU detection. Within local regions, however, many of these features are correlated.

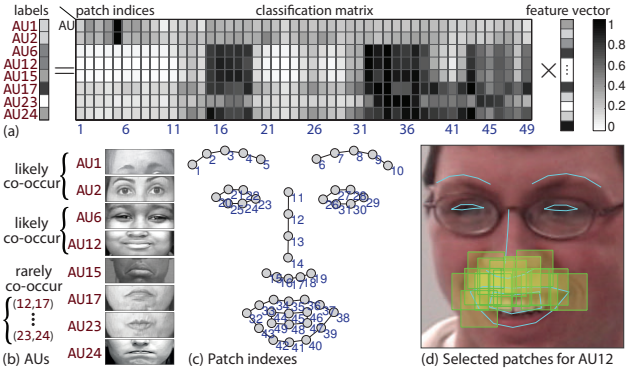


Figure 1. Joint patch and multi-label learning (JPML): (a) the learned classification matrix with consideration of positive and negative AU relations, (b) likely and rarely co-occurring AUs, (c) patch indexes, and (d) automatically selected patches for AU12.

We define local regions as patches centered around facial landmarks. By modeling features within local patches informed by FACS, it is possible to give greater weights to informative regions of interest and to reduce a large number of correlated features to achieve efficient learning. Zhong *et al*. [34] effectively applied patch learning to detect prototypic expressions (e.g., happy or sad). We apply patch learning to the more demanding problem of AU detection.

Similarly, just as features within patches have constraints, or correlation, AUs have constraints as well. AU 1 (inner-brow raise) increases the likelihood of AU 2 (outer-brow raise) and decreases the likelihood of AU 6 (cheek raiser). Multi-label learning builds upon this knowledge. Learning related AUs simultaneously improves learning in part by implicitly increasing the sample size for each AU. Recent efforts have explored AU relationships using Bayesian networks (BN) [25, 26] and dynamic Bayesian networks (DBN) [28]. Some developed generic domain knowledge to learn AU models without training data [15].

We address patch and multi-label learning with one stone. By taking both PL and ML into account, we model dependencies among both features and AUs. We explore two types of AU relations, termed *positive correlation* and *negative competition*, by statistically analyzing more than 350,000 samples from three varied datasets that include both posed and spontaneous facial behavior. The latter in-

cludes two- and three-person social contexts and a range of emotion inductions. Given such AU relations, we develop joint patch and multi-label learning (JPML) to simultaneously select a discriminative subset of patches and learn multi-AU classifiers. JPML leverages the structure in the classification matrix and AU labels, and naturally blends two tasks into one.

Fig. 1 illustrates the main idea. (a) shows a classification matrix in which columns correspond to patch indices and rows to individual AU classifiers; (b) shows likely and unlikely co-occurring AUs; (c) shows patch indices. (d) illustrates the patches selected by JPML, illustrating that JPML is able to finding a discriminative subset of patches to identify a target AU, in this case AU12 (oblique lip corner puller). In experiments, we will show that the joint processes of JPML are *mutually-beneficial* due to the complementary characteristics in the classification matrix.

2. Related Work

Automatic facial AU detection has been a vital research domain for objectively describing facial action related to emotion. See [1, 6, 9, 22] for comprehensive reviews. Our work closely follows recent efforts in patch learning and multi-label learning. Below we review each in turn.

Patch learning: Existing AU detection methods often perform *feature learning* to select a representative subset of raw features. Examples include AdaBoost [16], GentleBoost [27], and linear SVM [18]. However, as described in FACS [10], AUs relate to specific regions of human faces, *i.e.*, some facial regions are more important than others for recognizing specific AUs. If one seeks to detect brow raise (AUs 1 and 2), the eye and forehead regions are likely to be more informative than the jaw. Using domain knowledge, feature selection is sampled within subregions, or patches, of the face. Following this intuition, *patch learning* was proposed to model the region specificity to improve the performance of AU detection. Zhong *et al*. [34] divided a facial image into uniform patches, and then categorized these patches into common ones and specific ones according to basic expressions. Following a similar idea, Liu *et al*. [17] proposed to select common and specific patches corresponding to an expression pair (*e.g.*, happy-sadness). However, these patches were modeled implicitly and do not directly capture regional importance for certain AUs. Recently, Taheri *et al*. [24] used two-layer group sparse coding to encode AUs on predefined regions, and recovered facial expressions using sparsity in AU composition rules.

These patch learning approaches have been proved effective on posed expressions. However, the patch locations are pre-defined on a normalized template, and hence could fail to precisely capture the *specificity* of patches due to non-rigidity of human faces. Besides, it is unclear how AUs relations can be incorporated in these studies.

Multi-label learning: Existing research suggest the existence of strong AU correlations [15, 28]. For instance, AUs 6 and 12 are known co-occur in expressions of enjoyment and embarrassment. We can use such AU correlations to improve AU detection (*e.g.*, [5, 13, 18, 27]). To this end, Bayesian Networks (BN) [25, 26] and dynamic BN [28] have been used to exploit AU correlations. Other approaches exist, as well. Using generic domain knowledge, AU correlations can be modeled as a directional graph without training data [15]. In addition, a sparse multi-task model can be employed, assuming tasks are similar [32]. Without further research, it is unclear how these methods can best identify a discriminative subset of patches to improve AU detection. We propose a joint patch and multi-label learning (JPML) framework that simultaneously addresses patch- and multi-label learning for AU detection. These tasks prove mutually beneficial.

3. Joint Patch and Multi-label Learning (JPML)

3.1. Formulation

Let $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ be the training set with N instances and L AUs, where $\mathbf{x}_i \in \mathbb{R}^D$ is a feature vector from a facial image, and $\mathbf{y}_i \in \{+1, -1\}^L$ is an $L \times 1$ label vector which indicates a presence of the ℓ-th AU if the ℓ-th element $y_{i\ell} = +1$, and an absence of the ℓ-th AU if $y_{i\ell} = -1$ (see notation[1]). For notational convenience, we denote $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N] \in \mathbb{R}^{D\times N}$ as a data matrix, and $\mathcal{I}_\ell = \{i | y_{i\ell} = +1\}$ as an index set of instances that contain the ℓ-th AU. Our goal is to learn L linear classifiers in the matrix form $\mathbf{W} = [\mathbf{w}_1, ..., \mathbf{w}_L] \in \mathbb{R}^{D\times L}$ that enforces group-wise sparse feature selection (corresponding to the rows of $\mathbf{W}$) and label relations (corresponding to the columns of $\mathbf{W}$). We formulate JPML as an unconstrained optimization problem:

$$\min_{\mathbf{W}} \; L(\mathbf{W}, \mathcal{D}) + \alpha\Omega(\mathbf{W}) + \Psi(\mathbf{W}, \mathbf{X}), \qquad (1)$$

where $L(\mathbf{W}, \mathcal{D}) = \sum_{\ell=1}^{L} \sum_{i\in\mathcal{I}_\ell} \ln(1 + \exp(-y_{i\ell}\mathbf{w}_\ell^\top \mathbf{x}_i))$ is the logistic loss, $\Omega(\mathbf{W})$ is the *patch regularizer* that enforces sparse rows of $\mathbf{W}$ as *groups*, and $\Psi(\mathbf{W}, \mathbf{X})$ is a *relational regularizer* that constrains predictions on $\mathbf{X}$ with AU relations. Tuning parameters are α for $\Omega(\cdot)$ and (β_1, β_2) included in $\Psi(\cdot, \cdot)$. Problem (1) involves two tasks: identify a discriminative subset of patches for each AU (*patch learning*), and incorporate AU relations into model learning (*multi-label learning*). Below we detail each task in turn.

[1] Bold capital letters denote a matrix $\mathbf{X}$; bold lower-case letters denote a column vector $\mathbf{x}$. $\mathbf{x}_i$ the i-th column of the matrix $\mathbf{X}$. All non-bold letters represent scalars. X_{ij} denotes the scalar in the (i, j)-th entry of the matrix $\mathbf{X}$. x_j denotes the scalar in the jth element of $\mathbf{x}$. $\mathbf{1}_m \in \mathbb{R}^m$ is a vector of ones. $\mathbf{0}_{m\times n} \in \mathbb{R}^{m\times n}$ are matrices of zeros. $I(x)$ is an indicator function that returns 1 if the statement x is true, and 0 otherwise.

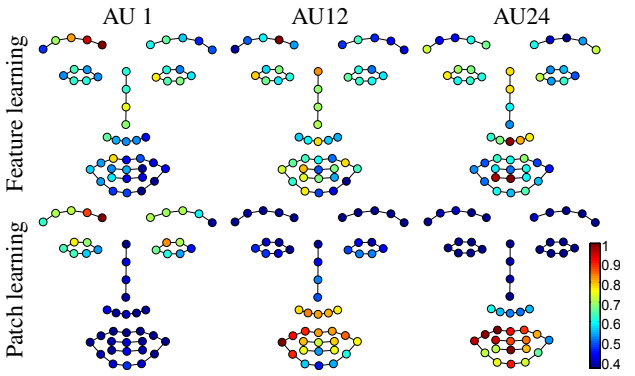


Figure 2. Patch importance between standard feature learning and our patch learning for AU1, 12 and 24 on CK+ dataset. Weights on each patch are computed as the norm of their classification vectors, and then normalized to [0,1].

3.2. Patch learning

The first task addresses *patch learning*. According to FACS [10], AUs are defined according to appearance changes at particular facial regions. Unlike standard feature learning methods that treats features separately [16, 19], patch learning constrains local dependencies in facial patches and gains better interpretation. Existing work select patches on uniformly distributed grid [17,24,34], while this paper exploits *landmark patches* that are centered at facial landmarks (as depicted in Fig. 1(c)). These landmark patches adapt better in real-world facial expression recognition scenario because of the non-rigidity of faces. In particular, we describe each patch using a 128-D SIFT descriptor. Each facial image is then represented as a 6272-D feature vector by concatenating SIFT descriptors of all landmarks.

To address the regional appearance changes on AUs, we define a group-wise sparsity on the classification matrix $\mathbf{W}$. Group sparsity learning aims to split variables into groups and then to select groups in sparsity. It has been shown to effectively recover joint sparsity across input dimensions, and successfully applied to computer vision (*e.g.*, [14, 31]). Given the structural nature of our problem, within each column of $\mathbf{W}$, we split every 128 values into non-overlapping groups, where each group corresponds to the SIFT features extracted from a particular patch. This grouping encourages a sparse selection of patches by jointly setting a group of rows to zero. In particular, Problem (1) reduces to:

$$\min_{\mathbf{W}} L(\mathbf{W}, \mathcal{D}) + \alpha\Omega(\mathbf{W}), \tag{2}$$

where $\Omega(\mathbf{W}) = \sum_{\ell=1}^{L}\sum_{p=1}^{49}\|\mathbf{w}_\ell^p\|_2$ is the *patch regularizer*, and $\mathbf{w}_\ell^p$ is the p-th group for the ℓ-th AU, *i.e.*, rows of $\mathbf{w}_\ell$ grouped by the patch p.

Patch importance: To validate the ability of maintaining the *specificity* of patches, we compare standard feature learning[2] (treat each feature independently) and our patch learning (treat features as groups), using the defined patch importance $\|\mathbf{w}_\ell^p\|_2$. As shown in Fig. 2, patch learning offers a better interpretation of important patches corresponding to three AU examples. For instance, patches around inner eyebrow contain higher importance for AU1; for AU24, patches around mouth (especially upper lips) are shown more important. Moreover, compared to previous work that manually defines a fixed region for AU12 (*e.g.*, [24,29]), our patch learning for AU12 automatically emphasizes not only upper lips (not lower lips), but also the patches around lower nose and slightly minor importance on the lower eyelid (corresponding to AU6). It can be seen that patch learning facilitates the *specificity* of relevant facial patches. Similar results could be obtained on other AUs and basic emotions.

[2] ℓ_1-regularized linear SVM [11] was used as feature learning.

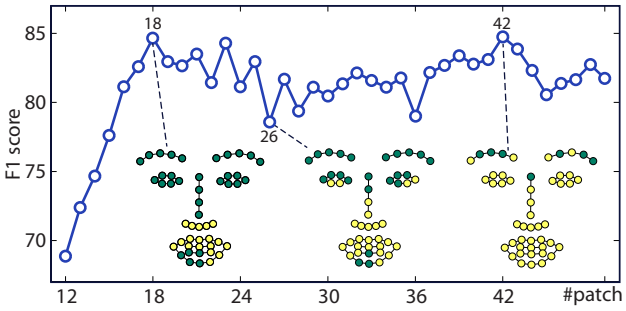


Figure 3. F1-Norm with respect to different #patches for AU12 on CK+ dataset. Three marked faces indicate the 18, 26 and 42 selected patches, which are depicted as light yellow circles.

#Patches versus performance: A natural question to ask is how the number of patches influences performance on AU detection. Intuitively, more patches should improve performance because more information is provided. To answer this question, we performed an experiment on AU12 using the CK+ dataset. Patches are selected in a descending order with respect to the patch importance. As shown in Fig. 3, the performance increases quickly until it hits the best performance with 18 patches, which associate with the zygomatic major in AU12 (upper lips and lower nose). When #patches become 25, patches on lower eyelid (associated with AU6) are included, showing that patches associated with AU6 are related to AU12. However, the performance drops slightly because not all patches carry useful information for a particular AU, coinciding with the findings [34]. Introducing more patches potentially include more noises that fluctuate the performance. Observing similar performance between #patches=18 and #patches=42, one can justify the importance of patch specificity, *i.e.*, only a subset of patches are discriminative for AU detection.

3.3. Multi-label learning

The next task is to exploit label relations for AU detection. Learning multiple related labels effectively increases the sample size for each class, and improves the prediction performance (*e.g.*, [3,30]). Despite the AU relations derived

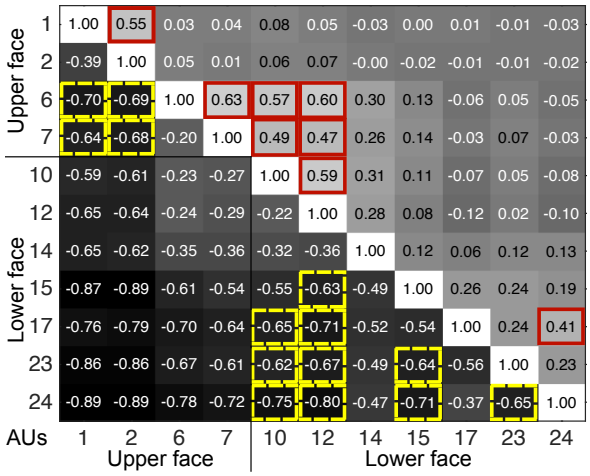


Figure 4. The relation matrix studies on more than 350,000 valid frames with AU labels. Red solid and dashed yellow rectangles, respectively, indicate the relations of *positive correlations* and *negative competitions* studied in this work.

from prior knowledge [15, 28], this section explores statistically the AU co-occurrence among more than 350,000 frames. Below we describe how we discover these relations, and how they can be incorporated into JPML.

Discover AU relations: We seek AU relations by statistically analyzing three datasets, CK+ [18], GFT [23] and BP4D [33], which contains 214 subjects and more than 350,000 valid frames with AU labels. The most frequently occurring AUs are used throughout this paper. Here, our goal is to discover likely and rarely co-occurring AUs.

Fig. 4 shows the relation matrix studied on the datasets. The (i, j)-th entry of the upper right matrix was computed as the coefficient correlation between the i-th and the j-th AU using ground truth labels; an entry of the lower left matrix was computed on the labels containing at least either the i-th or the j-th AU. One could interpret the upper matrix in Fig. 4 as a mutual relation of concurring AU pairs, and the lower matrix as an exclusive relation that one AU competes against another. After investigating this matrix with the FACS [10] and related studies [15, 28], we derive two types of AU relations, *positive correlation* and *negative competition*, as summarized in Table 1.

To discover these relations, we derive explicit rules as follows. AUs with over moderate positive correlations, *i.e.*, correlation coefficient ≥ 0.40, are assigned as *positive correlations*, *e.g.*, AUs (6, 12) co-occur frequently to describe a Duchenne smile. AUs with large negative correlations, *i.e.*, correlation coefficient ≤ 0.60, are selected as *negative competitions*, implying these AUs compete against each other and thus avoid occurring at the same time, *e.g.*, AUs (12, 15) have negative influences on each other (coincide with the findings in [15]). Note that, for the lower matrix, we exclude the consideration of relations between upper face and lower face AUs, because their facial muscles function

Table 1. AU relations discovered and used in this study

AU relations	**AU groups**
Positive correlation	(1,2), (6,7), (6,10), (7,10), (6,12), (7,12), (10,12), (17,24)
Negative competition	(1,6), (1,7), (2,6), (2,7), (10,17), (10,23), (10,24), (12,15), (12,17), (12,23), (12,24), (15,23), (15,24), (23,24)

separately and thus do not compete against each other. In addition, one can observe that the absolute values of lower matrix are much larger than the upper ones, providing another evidence that out of thousands of AU combinations, most rarely co-occur, coinciding with [24].

Incorporate AU relations into JPML: Denote the set of AU pairs with positive correlations and with negative competitions as $\mathcal{P}$ and $\mathcal{N}$, respectively. For instance, (1,2) and (6,12) are in $\mathcal{P}$; (15,23), (15,24), and (23,24) are in $\mathcal{N}$. To incorporate the AU relations discovered above, we introduce the *relational regularizer* as:

$$\Psi(\mathbf{W}, \mathbf{X}) = \beta_1 PC(\mathbf{W}, \mathbf{X}, \mathcal{P}) + \beta_2 NC(\mathbf{W}, \mathbf{X}, \mathcal{N}), \quad (3)$$

where β_1 and β_2 are tradeoff coefficients. $PC(\mathbf{W}, \mathbf{X}, \mathcal{P})$ captures the AU relations of positive correlations:

$$PC(\mathbf{W}, \mathbf{X}, \mathcal{P}) = \frac{1}{2} \sum_{(i,j)\in\mathcal{P}} \gamma_{ij} \|\mathbf{w}_i^\top \mathbf{X} - \mathbf{w}_j^\top \mathbf{X}\|_2^2, \quad (4)$$

where γ_{ij} is a pre-defined similarity score that determines how similar two predictions $\mathbf{w}_i^\top \mathbf{X}$ and $\mathbf{w}_j^\top \mathbf{X}$ are. The larger γ_{ij} is, the more similar predictions are for the i-th and the j-th AUs in $\mathcal{P}$ ($\gamma_{ij} = 2000$ in our experiments). The intuition behind this regularizer is that positively correlated AUs implies similar predictions. $NC(\mathbf{W}, \mathbf{X}, \mathcal{N})$ is defined in analogy to exclusive lasso [35]: $NC(\mathbf{W}, \mathbf{X}, \mathcal{N})q = \sum_{i=1}^{N} \sum_{n=1}^{|\mathcal{N}|} \left(\sum_{j\in\mathcal{N}_n} |\mathbf{w}_j^\top \mathbf{x}_i| \right)^2$, where $\mathcal{N}_i$ indicates the i-th element in $\mathcal{N}$, and $|\mathcal{N}| = 14$ in our case (as shown in Table 1). For example, $\mathcal{N}_1$ is the AU pair (1,6) with negative competition. Because the ℓ_1 norm tends to achieve a sparse solution, if one classifier predicts AU1 in the group $\mathcal{N}_1$, the AU6 classifier tends to generate small prediction values. In this way, we are able to introduce competitions among the predictions within the same negative group. As a result, we solve for the multi-label learning task of JPML:

$$\min_{\mathbf{W}} \; L(\mathbf{W}, \mathcal{D}) + \Psi(\mathbf{W}, \mathbf{X}). \quad (5)$$

We detail our algorithm to solve JPML as follows.

3.4. Algorithm

Because $\Omega(\mathbf{W})$ and $\Psi(\mathbf{W}, \mathbf{X})$ constrain on $\mathbf{W}$ differently, Problem (1) cannot be solved directly. We rewrite Problem (1) by introducing auxiliary variables $\mathbf{W}_1, \mathbf{W}_2$.,

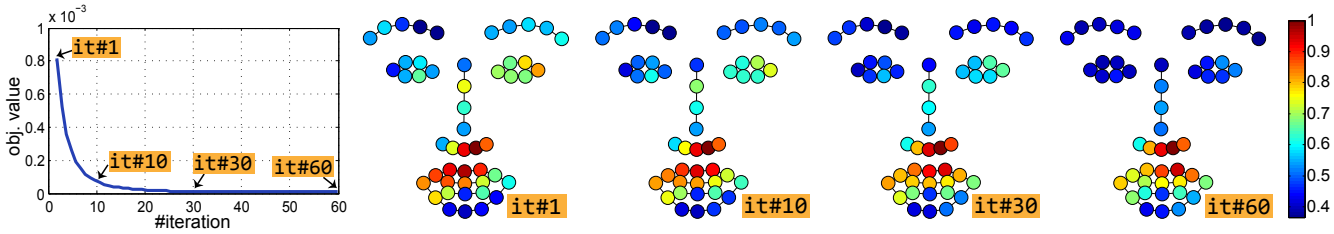


Figure 5. Illustration of convergence curve on learning active patches on AU12 with algorithm PL. While the iterations proceed, PL identifies the regions for AU12 (lip corner puller) with better specificity.

Algorithm 1 Patch learning (PL)

Input: Training data $\mathcal{D}=\{(\mathbf{x}_i,\mathbf{y}_i)\}_{i=1}^N$, ML matrix $\mathbf{W}_2$, Lagrange multiplier of ADMM ρ and $\mathbf{U}$, learning rate η_1, and penalty parameter α.

Output: PL matrix $\mathbf{W}_1\in\mathbb{R}^{D\times L}$ with sparse groups of rows.

for $\ell=1,\ldots,L$ **do**
  $\mathbf{w}_{1\ell}^{(0)}=\frac{1}{D}\mathbf{1}_D, \mathbf{v}^{(0)}=\frac{1}{D}\mathbf{1}_D, a^{(0)}=1, t=0$; // Initialization
  while not convergence **do**
    $\mathbf{z}^{(t)}=\mathbf{v}^{(t)}-\eta_1(\nabla L(\mathbf{w}_{1\ell}^{(t)},\mathcal{D})+\mathbf{u}_\ell^{(t)}+\rho(\mathbf{w}_{1\ell}^{(t)}-\mathbf{w}_{2\ell}^{(t)}))$;
    for $p=1,\ldots,49$ **do**
      $\mathbf{w}_{1\ell}^{p(t+1)}=I(\|\mathbf{z}^{p(t)}\|_2>\alpha)(1-\frac{\alpha}{\|\mathbf{z}^{p(t)}\|_2})\mathbf{z}^{p(t)}$;
      // $\mathbf{w}_{1\ell}^p$ is the p-th patch within the ℓ-th column of $\mathbf{W}_1$
    end for
    $a^{(t+1)}=\frac{2}{t+1}$;
    $\mathbf{v}^{(t+1)}=\mathbf{w}_{1\ell}^{(t+1)}+(\frac{1-a^{(t)}}{a^{(t)}}a^{(t+1)})(\mathbf{w}_{1\ell}^{(t+1)}-\mathbf{w}_{1\ell}^{(t)})$;
    $t=t+1$;
  end while
end for

and then jointly optimize $\mathbf{W}_1$ and $\mathbf{W}_2$ using ADMM [2]:

$$\min_{\mathbf{W}_1,\mathbf{W}_2} L(\mathbf{W}_1,\mathcal{D})+\alpha\Omega(\mathbf{W}_1)+\Psi(\mathbf{W}_2,\mathbf{X})+\frac{\rho}{2}\|\mathbf{W}_1-\mathbf{W}_2\|_F^2 \quad \text{s.t.}\,\mathbf{W}_1=\mathbf{W}_2. \tag{6}$$

The augmented Lagrangian can be written as:

$$\mathcal{L}_\rho(\mathbf{W}_1,\mathbf{W}_2,\mathbf{U})=L(\mathbf{W}_1,\mathcal{D})+\alpha\Omega(\mathbf{W}_1)+\Psi(\mathbf{W}_2,\mathbf{X}) +\langle\mathbf{U},\mathbf{W}_1-\mathbf{W}_2\rangle+\frac{\rho}{2}\|\mathbf{W}_1-\mathbf{W}_2\|_F^2. \tag{7}$$

ADMM consists of three updates:

$$\mathbf{W}_1^{(k+1)}=\min_{\mathbf{W}_1}\mathcal{L}_\rho(\mathbf{W}_1,\mathbf{W}_2^{(k)},\mathbf{U}^{(k)}), \tag{8}$$

$$\mathbf{W}_2^{(k+1)}=\min_{\mathbf{W}_2}\mathcal{L}_\rho(\mathbf{W}_1^{(k+1)},\mathbf{W}_2,\mathbf{U}^{(k)}), \tag{9}$$

$$\mathbf{U}^{(k+1)}=\mathbf{U}^{(k)}+\rho(\mathbf{W}_1^{(k+1)}-\mathbf{W}_2^{(k+1)}). \tag{10}$$

Solving (8) involves the patch regularizer $\Omega(\mathbf{W}_1)$ and the augmented terms in $\mathcal{L}_p$. Because solving for $\mathbf{W}_1$ with $L_{2,1}$ norm is a non-smooth problem, here we use the accelerated gradient method [4] to decompose $L_{2,1}$ norm into 49 sub-problems. Algo. 1 summarizes the detailed procedure. The convergence condition in the algorithm is $\|\mathbf{w}^{(t+1)}-\mathbf{w}^{(t)}\|_2\le\delta$ ($\delta=10^{-5}$ in our case).

Algorithm 2 Multi-label learning (ML)

Input: Training data $\mathcal{D}=\{(\mathbf{x}_i,\mathbf{y}_i)\}_{i=1}^N$, PL matrix $\mathbf{W}_1$, Lagrange multiplier of ADMM ρ and $\mathbf{U}$, learning rate η_2, penalty parameter β_2, and accuracy control parameter μ.

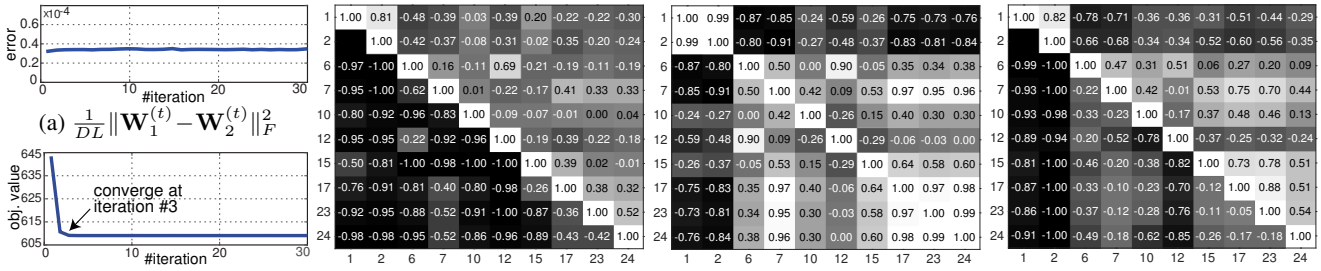
Output: ML matrix $\mathbf{W}_2\in\mathbb{R}^{D\times L}$.

$\mathbf{W}_2^{(0)}=\frac{1}{D}\mathbf{1}_{D\times L}, \mathbf{V}^{(0)}=\frac{1}{D}\mathbf{1}_{D\times L}, a^{(0)}=1, t=0$; // Init.
while not convergence **do**
  $\mathbf{U}^{(t)}=(1-a^{(t)})\mathbf{W}_2^{(t)}+a^{(t)}\mathbf{V}^{(t)}$;
  $\mathbf{H}_\mu=\mathbf{0}_{L\times D}$;
  for $i=1,\ldots,N$ **do**
    $\mathbf{z}_i=\min(1,\max(-1,\frac{\mathbf{U}^{(t)\top}\mathbf{x}_i}{\mu}))$;
    $q_i=\mathbf{z}_n^\top\mathbf{U}^{(t)\top}\mathbf{x}_i-\frac{\mu}{2}\|\mathbf{z}_i\|_2^2$;
    $\mathbf{H}_\mu=\mathbf{H}_\mu+q_i(\mathbf{z}_i\mathbf{x}_i^\top)$;
  end for
  $\mathbf{V}^{(t+1)}=\mathbf{V}^{(t)}-\frac{1}{\eta_2}(\mathbf{H}_\mu^\top-\mathbf{u}+\rho(\mathbf{W}_1-\mathbf{U}^{(t)}))+\nabla PC(\mathbf{U}^{(t)})$;
  $\mathbf{W}_2^{(t+1)}=(1-a^{(t)})\mathbf{W}_2^{(t)}+a^{(t)}\mathbf{V}^{(t+1)}$;
  $a^{(t+1)}=\frac{2}{t+1}$;
  $t=t+1$;
end while

Fig. 5 illustrates the convergence process of PL on AU12. While the number of iteration increases, PL converges to a subset of patches that preserves better specificity. On iteration #1, many patches are selected and thus remain an ambiguous representation. From iteration #10 to #30, patches associated with AU12 are strengthen but still involve unrelated regions such as eyes. PL converges at it#60, revealing the discriminative patches around lower nostril wing and upper mouth, the regions that zygomaticus major muscle triggers for AU12.

Solving (9) involves the relational regularizer $\Psi(\mathbf{W}_2,\mathbf{X})$ and the augmented terms in $\mathcal{L}_p$. For $\Psi(\cdot,\cdot)$, the positive correlation $PC(\mathbf{W}_2,\mathbf{X},\mathcal{P})$ is smooth in $\mathbf{W}_2$, but the negative competition $NC(\mathbf{W}_2,\mathbf{X},\mathcal{N})$ is not. Here we adopt Nesterov's approximation [21] to smooth the objective. Given a training sample $\mathbf{x}_i$ and its negative relation $\mathcal{N}_i$, we denote $\mathbf{W}_{\mathcal{N}_i}$ as a $D\times|\mathcal{N}_i|$ matrix where each column contains $\mathbf{w}_j$ and $j\in\mathcal{N}_i$. Let $\|\mathbf{W}_{\mathcal{N}_i}^\top\mathbf{x}_i\|_1=\sum_{j\in\mathcal{N}_i}|\mathbf{w}_j^\top\mathbf{x}_i|$, we can write its dual norm as $\|\mathbf{W}_{\mathcal{N}_i}^\top\mathbf{x}_i\|_1=\max_{|\mathbf{z}|\le1}\langle\mathbf{W}_{\mathcal{N}_i}^\top\mathbf{x}_i,\mathbf{z}\rangle$, and smooth $NC(\mathbf{W}_2,\mathbf{X},\mathcal{N})$ following [21]. See Algo. 2.

JPML is optimized by iterating patch learning (Algo. 1) and multi-label learning (Algo. 2). Because the ADMM



(a) $\frac{1}{DL}\|\mathbf{W}_1^{(t)} - \mathbf{W}_2^{(t)}\|_F^2$ (b) Obj. value of Eq. (7) (c) Relation matrix in CK+ (d) Patch learning (e) JPML

Figure 6. Illustration of JPML on the CK+ dataset: (a) $\frac{1}{DL}\|\mathbf{W}_1^{(t)} - \mathbf{W}_2^{(t)}\|_F^2$ v.s. #iteration, (b) objective value in (7) v.s. #iteration, (c) ground truth relation matrix (correlation coefficients between ground truth AU labels), (d) relation matrix at the initialization step (with patch learning only), and (e) relation matrix computed by predictions of JPML. The difference of correlation coefficient between (c) and (d) is 0.51, and that between (c) and (e) is 0.15, showing that JPML helps preserve the relations between AUs.

form in (7) is bi-convex, it is guaranteed to converge to a critical point. Fig. 6 shows the convergence process of JPML. In training, the maximum iteration is set as 30, while JPML typically converges within 5 iterations. As can be seen in (a), for each iteration of PL and ML, JPML manages to keep the averaged error between $\mathbf{W}_1^{(t)}$ and $\mathbf{W}_2^{(t)}$ as low as 10^{-5}. By adding *positive correlations* and *negative competitions* into patch learning, much more accurate correlations closed to ground truth can be learned. In quantities, the distance between predictions and ground truth decreased 3.4 times, as shown in Fig. 6(d) and (e). Note that the entry of AUs (1,2) in Fig. 6(c)∼(e) is empty because in CK+ AUs (1,2) always co-occur, leading to a zero variance during the computation of correlation coefficient.

4. Experiments

4.1. Settings

Datasets: We evaluated the effectiveness of JPML in three datasets that include both posed and spontaneous facial behavior in varied contexts. Each database had been FACS coded by well-experienced coders. Inter-observer agreement in each was quantified using coefficient kappa, which control for chance agreement between coders, and it was maintained at a kappa of 0.80 or higher, which represents high inter-observer agreement.

(1) CK+ [18] is a leading testbed for facial expression analysis. It consists of 593 sequences of posed facial actions from 123 subjects. The first and the last frames of each sequence were selected as negative and positive samples, respectively. In all, 593 images with 10 AUs were used.

(2) GFT [23] consists of 720 participants recorded during group-formation tasks. Previously unacquainted participants sat together in groups of 3 at a round table for 30 minutes while getting to know each other. We used 2 minutes of video from 50 participants. For each participant, we randomly sampled 100 positive frames and 200 negative frames for training purposes.

(3) BP4D [33] contains 2D/3D videos of spontaneous facial expressions in young adults during various emotion inductions while interacting with an experimenter. We used 328 2D videos from 41 participants. For each video, we randomly sampled 50 positive frames and 100 negative frames for training purpose.

Because severely skewed base rates attenuate estimates of classifier performance, only AU occurring more than 3% to 5% of the time were included for analysis. Across datasets, 10 to 11 AU met this criterion. Even though AU with very low base rates were omitted, skew nevertheless varied considerably. To control for the effects of skew on AU detection, test statistics were normalized for skew using the procedure of [12]. By normalizing for skew we were able to reliably compare results within and between datasets. Table 2 summarizes the skew factor defined as the ratio of the number of negative samples to the number of positive ones.

Pre-processing: IntraFace [7] was used to track 49 facial landmarks. Tracked landmarks were registered to a reference face using similarity transform. Appearance features were extracted using SIFT descriptor [36] at frame level, resulting in 49×128-D features for each image. To take full advantage of the datasets, we divided GFT and BP4D into 10 splits of independent participants. Because CK+ only contains 593 images, 5 splits were adopted.

Evaluation metrics: To report objective results, we used two metrics to compare performance, F1-Norm (frame-based) and F1-Event (segment-based). F1-Norm [12] is computed as the normalized F1 score with a skew factor: F1-Norm $= \frac{2s \cdot R \cdot P}{s \cdot R + P}$, where R is recall, P is precision, and s is the skew factor. F1-Norm skew-normalizes the standard F1 metric and enables comparison both within and between datasets. On the other hand, F1-Event [8] serves as a segment-based metric defined as the harmonic mean between event-based recall ER and event-based precision EP: F1-Event $= \frac{2 \cdot ER \cdot EP}{ER + EP}$. For each method, we computed the averaged metric over all AUs (denoted as *AA.*), and averaged over only the AUs with relationships (denoted as *AR.*).

Comparative methods: To investigate the benefits of

Table 2. Skew on each AU within different datasets

AU	1	2	6	7	10	12	14	15	17	23	24
CK+	1.5	2.3	2.4	3.1	20.8	3.1	9.2	8.9	1.8	6.6	6.6
GFT	10.1	8.3	2.1	1.5	1.5	2.0	0.6	9.7	2.7	4.9	8.4
BP4D	3.8	4.9	1.2	0.8	0.7	0.8	1.1	4.9	1.9	5.0	5.5

JPML, we compared it with methods that omit patch- and multi-label learning and with approaches that use patch- or multi-label learning but not an integration of both.

For baseline without PL or ML, we trained Linear SVMs (LSVM) [11] on individual AU. As a baseline for *feature learning*, we used L1-regularized logistic regression (LL1) [11]. All use features without considering patches.

For PL, we used several patch selection methods. These were self-defined patches (similar to [5, 36]) with binary SVMs, termed as SP-SVM, in comparison to our automatic patch selection. Patches were defined according to FACS and patch indexes in Fig. 1(c): landmarks #1∼#10 are assigned to AUs 1, 2, and 7; #11∼#30 for AU6; #11∼#19 for AUs 11 and 14, #32∼#49 for all AUs around lips. Patches on eyebrows were selected for training classifiers on AUs 1, 2 and 7; patches on eyes and nose for AU 6; patches around nose for AUs 11 and 14; patches around lips for all AUs around mouth. In addition, we compared two state-of-the-art patch learning methods, Structure Preserving Sparse Decomposition (SPSD) [24] and Active Patch Learning (APL) [34]. For SPSD, because GFT and BP4D do not contain expressions labels, we used one layer to learn AU dictionary, and K-SVD [20] to learn AU atoms on fixed patches. Note that the original APL [34] was defined on emotion bases using uniform segmentation on face images. In our experiments, we implemented APL using patches centered at landmarks and algorithm in Algo. 1.

For ML, we compare with MT-MKL [32] using RBF and polynomial kernels with the implementation provided by the authors. Because MT-MKL involves computing multiple kernel matrices, it is computationally prohibitive for large datasets such as GFT and BP4D, and was carried out only on CK+. Following [32], we employed 3 AU groups overlapped with this study: AUs (1,2), (6,12), and (15,17). According to parameters in Algos. 1 and 2, α is cross-validated within $\{10^{-3}, 10^{-4}, 10^{-5}\}$, $\eta_1 = 10^{-4}$, $\gamma = 2000$, $\mu = 10^{-4}$, $\eta_2 = 2000$, $\beta_1 = 10^{-3}$, and $\beta_2 = 10^{-4}$.

4.2. Results

Tables 3∼5 show the results on CK+, GFT, and BP4D, respectively. AUs *without* relationships are underlined. We excluded these AUs for ML and JPML and denoted theirs results as "–". For CK+, because each video starts from a neutral face to a particular peak expression, we evaluated with only F1-Norm. For GFT and BP4D consisting of spontaneous videos, we used both F1-Norm and F1-Event to capture the imbalance nature of AU detection and the ability to preserve temporal consistency. Below we discuss the results from three perspectives: patch Learning, multi-label learning and the proposed joint framework JPML.

Table 3. Comparisons on the CK+ dataset. Bracketed numbers stand for the best performance; bold numbers for the second best.

	F1-Norm							
AU	SP-SVM	SPSD	LSVM	LL1	MT-MKL	ML	APL	JPML
1	61.8	44.4	85.8	83.4	73.0	**89.0**	86.4	[**90.0**]
2	63.9	47.9	90.9	87.4	87.8	**92.7**	86.6	[**93.0**]
6	61.7	34.2	**75.3**	[**76.2**]	61.9	70.7	70.5	74.2
7	60.0	50.8	[**70.8**]	**70.0**	–	61.6	62.8	66.7
12	65.5	47.4	[**80.7**]	80.0	73.3	75.2	76.6	[**80.7**]
14	66.3	59.9	**67.7**	**67.7**	–	–	[**69.5**]	–
15	65.8	53.4	67.5	66.7	67.8	61.9	**79.2**	**73.6**
17	60.4	62.2	**80.5**	**80.5**	68.3	80.3	80.0	[**83.5**]
23	66.2	65.0	69.3	69.8	–	69.7	**83.5**	**74.3**
24	68.3	65.8	71.1	**71.4**	–	67.5	**75.9**	65.9
AA.	64.0	52.3	**76.0**	75.3	–	–	[**77.1**]	–
AR.	63.7	51.5	76.9	76.2	72.0	74.3	**77.0**	[**78.0**]

Patch learning: This paragraph attempts to answer the question: does APL help improve performance compared to standard feature learning and patch learning methods? Out of three datasets, we evaluated 32 AUs with F1-Norm, and 22 AUs with F1-event. In general, APL outperforms features learning (LL1 and LSVM) in 26/32 AUs for F1-Norm, and 14/22 AUs for F1-event. Compared to patch learning approaches (SP-SVM and SPSD) that use uniformly distributed patches, APL outperforms in 30/32 AUs with F1-Norm and 17/22 in F1-event. One explanation is that our APL uses patches around facial landmarks, and thus better adapts to appearance changes on spontaneous expressions. In particular, as can be seen in Tables 3∼5, APL performs more effectively when applied to lower face AUs, which typically involves larger motions on mouth regions. In summary, we justify that APL is more reasonable than standard feature learning and patch learning with fixed patches.

Multi-label learning: This paragraph discusses the benefits of considering relations between AU labels using multi-label learning. Closest to our work is MT-MKL that assumes classifiers within the same AU group behave similarly. On the contrary, our ML (Sec. 3.3) considers positive correlation as well as negative competition on labels (instead of classifiers), and thus more naturally fits the problem in hand. In Table 3, averaging F1-Norm over the 6 AUs we implemented for MT-MKL, ML outperforms against MT-MKL by 8.8%. In Tables 4 and 5, we have seen that ML consistently outperforms standard binary classifiers (LL1, LSVM, SPSD and SP-SVM), showing that relations between AU labels are essential to assist AU detection.

JPML: APL and ML alone have shown good performance over three datasets. This paragraph focuses on the

Table 4. Comparisons on the GFT dataset. Bracketed numbers indicate the best performance; bold numbers indicate the second best.

	F1 Norm							F1 Event						
AU	SPSVM	SPSD	LSVM	LL1	ML	APL	JPML	SPSVM	SPSD	LSVM	LL1	ML	APL	JPML
1	29.9	33.0	53.0	52.0	[**66.7**]	44.1	**58.0**	**17.8**	12.2	[**20.6**]	17.8	11.5	11.5	15.9
2	60.2	34.7	51.3	45.1	[**64.4**]	43.6	**63.2**	[**21.2**]	12.9	19.6	16.8	12.5	**16.6**	15.0
6	[**77.2**]	34.8	74.7	75.2	57.3	**77.2**	[**79.6**]	46.6	21.6	33.2	42.3	25.5	**50.3**	[**50.8**]
7	56.5	40.3	**72.7**	70.5	67.6	[**73.6**]	[**73.6**]	41.3	25.3	38.2	34.4	34.3	**47.9**	[**54.7**]
10	74.6	41.8	75.8	**77.5**	–	[**78.6**]	–	**45.6**	30.7	41.2	37.9	–	**50.2**	–
12	77.1	76.2	79.2	80.2	67.1	**81.3**	[**84.1**]	47.9	**48.6**	47.9	48.4	15.3	[**53.6**]	46.7
14	64.1	**68.9**	68.5	[**70.4**]	–	66.7	–	42.1	49.0	42.1	**55.0**	–	[**60.6**]	–
15	47.2	30.1	45.8	65.3	66.3	[**67.1**]	**66.2**	16.4	10.6	**39.1**	[**39.7**]	17.8	18.9	**37.9**
17	51.8	32.8	47.6	46.8	67.1	[**74.5**]	**72.0**	33.8	22.9	38.3	**38.9**	27.1	[**48.7**]	38.8
23	49.7	35.9	38.8	43.5	[**66.9**]	**63.9**	60.0	25.9	18.0	**35.4**	28.4	28.6	35.0	[**37.6**]
24	51.1	35.3	56.6	59.2	67.1	**79.0**	[**79.3**]	18.7	12.9	**27.3**	25.0	26.7	19.2	[**35.5**]
AA.	56.5	42.3	59.8	55.4	–	[**67.1**]	–	31.1	23.4	34.2	34.6	–	[**36.3**]	–
AR.	53.6	39.4	57.0	51.3	65.6	**65.9**	[**70.7**]	38.3	19.7	32.5	32.0	22.1	**32.7**	[**37.0**]

Table 5. Comparisons on the BP4D dataset. Bracketed numbers indicate the best performance; bold numbers indicate the second best.

	F1 Norm							F1 Event						
AU	SPSVM	SPSD	LSVM	LL1	ML	APL	JPML	SPSVM	SPSD	LSVM	LL1	ML	APL	JPML
1	22.9	27.6	40.6	35.6	[**58.6**]	**56.0**	55.5	9.5	10.6	13.0	11.7	14.9	**17.1**	[**17.5**]
2	15.8	15.8	32.1	24.1	56.9	**60.2**	[**62.7**]	7.6	7.6	11.5	10.3	14.0	**16.0**	[**17.2**]
6	45.5	54.6	59.4	**75.2**	62.9	75.0	[**75.7**]	21.9	27.9	17.2	21.1	15.6	[**32.7**]	**30.0**
7	44.1	56.0	55.7	[**70.5**]	66.7	64.3	**66.7**	22.9	**30.5**	20.5	23.6	17.1	[**33.7**]	26.3
10	50.1	55.6	63.0	[**74.3**]	–	**72.9**	–	29.7	32.8	22.0	**34.1**	–	**41.0**	–
12	46.5	54.9	62.5	82.0	67.1	[**82.3**]	**81.4**	28.4	30.6	23.4	25.0	20.5	[**41.3**]	**31.6**
14	44.2	52.7	51.5	61.2	–	**66.0**	–	19.3	28.3	23.5	29.3	–	**29.8**	–
15	13.2	40.5	49.6	56.3	66.0	[**68.4**]	**65.9**	**23.4**	22.9	23.9	18.6	20.4	13.1	[**30.1**]
17	42.3	46.9	40.3	63.4	66.7	[**69.2**]	**65.3**	19.3	21.9	21.2	25.6	20.8	[**33.5**]	**29.4**
23	11.3	23.9	42.1	57.2	67.1	[**68.0**]	**65.2**	**19.4**	19.6	[**21.8**]	19.0	20.6	16.2	[**27.7**]
24	7.3	47.3	21.3	69.5	66.7	[**78.1**]	**77.3**	17.7	18.4	19.0	[**23.1**]	20.4	13.2	[**26.4**]
AA.	29.3	42.1	47.1	59.5	–	[**68.7**]	–	18.9	21.8	19.7	23.1	–	[**24.6**]	–
AR.	25.3	39.4	44.8	57.7	64.3	[**68.5**]	**68.4**	15.9	19.9	19.0	21.1	18.3	**22.2**	[**26.2**]

discussion of JPML that jointly considers patch selection and AU relations. In all, JPML achieves the best or second best for 22/27 AUs in F1-Norm and for 12/18 AUs for F1-event. In Table 3, JPML performs the best for AUs (1,2,12,15), and improves about 1.3% and 5.0% than APL and ML respectively for F1-norm. It improves more than 7.3% and 7.8% for F1-Norm, and 13% and 67% for F1-event than APL and ML respectively. In Tables 4 and 5, as more spontaneous expression are involved, the improvement becomes more obvious. Since the ratio of training and test samples in BP4D is a little small in this paper and samples in BP4D is much more complex than GFT, the results in Table 5 is smaller than ones in Table 4 in average. In all, JPML method achieved the highest overall scores in five comparisons on three datasets. In BP4D, APL is slightly higher than JPML. In no cases, the other approaches match or exceed APL and JPML. This suggests that our patch-based approach is more powerful, and further boost the performance with additional ML. In addition, there are some interesting observations in our results. JPML yields better improvement in AUs with larger skew (*e.g.*, AU1 and AU2 in GFT and BP4D), as shown in Table 2. To summarize, JPML validates the effectiveness of jointly learning the patches and AU relations, showing that iterating the ML and the APL process is beneficial.

5. Conclusion

This paper proposes a joint patch and multi-label learning (JPML) for facial AU detection. Active patches for each AU are selected more *specificity* by group sparsity learning. Jointly with patch learning, *positive correlations* and *negative competitions* among AUs are introduced to model a discriminative multi-label classifier. Compared with patch learning based and multi-label learning based algorithms separately, JPML obtained the best predictions across three datasets. According to the conclusion of results in experiments, imbalance data learning and video-based learning algorithm should be studied in the future work.

Acknowledgments

Research reported in this paper was supported in part by US National Institutes of Health under Award Number MH096951 and National Science Foundation under grant RI-1116583. K. Zhao and H. Zhang are supported by Natural Science Foundation of China under grant 61273217, 61175011, and 61402047.

References

[1] M. S. Bartlett, G. Littlewort, C. Lainscsek, I. Fasel, and J. Movellan. Machine learning methods for fully automatic recognition of facial expressions and facial actions. In *Systems, Man and Cybernetics*, 2004.

[2] 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.

[3] R. S. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino. Matrix completion for multi-label image classification. In *Advances in Neural Information Processing Systems*, 2011.

[4] X. Chen, W. Pan, J. T. Kwok, and J. G. Carbonell. Accelerated gradient method for multi-task sparse learning problem. In *ICDM*, 2009.

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

[6] J. F. Cohn and F. De la Torre. The oxford handbook of affective computing. *Automated Face Analysis for Affective Computing*, 2014.

[7] F. De la Torre, W.-S. Chu, X. Xiong, X. Ding, and J. F. Cohn. Intraface. In *AFGR*, 2015.

[8] 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.

[9] S. Du, Y. Tao, and A. M. Martinez. Compound facial expressions of emotion. *Proceedings of the National Academy of Sciences*, 111(15):E1454–E1462, 2014.

[10] P. Ekman, W. Friesen, and J. C. Hager. Facial action coding system. *A Human Face*, 2002.

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

[12] L. A. Jeni, J. F. Cohn, and F. De La Torre. Facing imbalanced data–recommendations for the use of performance metrics. In *Affective Computing and Intelligent Interaction*, 2013.

[13] S. Koelstra, M. Pantic, and I. Patras. A dynamic texture-based approach to recognition of facial actions and their temporal models. *TPAMI*, 32(11):1940–1954, 2010.

[14] L.-J. Li, H. Su, L. Fei-Fei, and E. Xing. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, 2010.

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

[16] G. Littlewort, M. S. Bartlett, I. Fasel, J. Susskind, and J. Movellan. Dynamics of facial expression extracted automatically from video. *Image and Vision Computing*, 24(6):615–625, 2006.

[17] P. Liu, J. T. Zhou, I. W.-H. Tsang, Z. Meng, S. Han, and Y. Tong. Feature disentangling machine-a novel approach of feature selection and disentangling in facial expression analysis. In *ECCV*, 2014.

[18] 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 *CVPRW*, 2010.

[19] S. Lucey, A. B. Ashraf, and J. Cohn. Investigating spontaneous facial action recognition through aam representations of the face. *Face recognition*, 32(11):275–286, 2010.

[20] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *ICML*, 2009.

[21] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[22] M. Pantic and L. J. M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *Pattern Analysis and Machine Intelligence*, 22(12):1424–1445, 2000.

[23] M. A. Sayette, K. G. Creswell, J. D. Dimoff, C. E. Fairbairn, J. F. Cohn, B. W. Heckman, T. R. Kirchner, J. M. Levine, and R. L. Moreland. Alcohol and group formation a multimodal investigation of the effects of alcohol on emotion and social bonding. *Psychological science*, 2012.

[24] S. Taheri, Q. Qiu, and R. Chellappa. Structure-preserving sparse decomposition for facial expression analysis. *TIP*, 23(8):3590–3603, Aug 2014.

[25] Y. Tong and Q. Ji. Learning bayesian networks with qualitative constraints. In *CVPR*, 2008.

[26] Y. Tong, W. Liao, and Q. Ji. Inferring facial action units with causal relations. In *CVPR*, 2006.

[27] M. Valstar and M. Pantic. Fully automatic facial action unit detection and temporal analysis. In *CVPRW*, 2006.

[28] Z. Wang, Y. Li, S. Wang, and Q. Ji. Capturing global semantic relationships for facial action unit recognition. In *ICCV*, 2013.

[29] J. Whitehill, G. Littlewort, I. Fasel, M. Bartlett, and J. Movellan. Toward practical smile detection. *TPAMI*, 31(11):2106–2111, 2009.

[30] Z.-J. Zha, X.-S. Hua, T. Mei, J. Wang, G.-J. Qi, and Z. Wang. Joint multi-label multi-instance learning for image classification. In *CVPR*, 2008.

[31] S. Zhang, J. Huang, Y. Huang, Y. Yu, H. Li, and D. N. Metaxas. Automatic image annotation using group sparsity. In *CVPR*, 2010.

[32] X. Zhang, M. H. Mahoor, S. M. Mavadati, and J. F. Cohn. A lp-norm mtmkl framework for simultaneous detection of multiple facial action units. In *WACV*, 2014.

[33] 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 Workshop*, 2013.

[34] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, and D. N. Metaxas. Learning active facial patches for expression analysis. In *CVPR*, 2012.

[35] Y. Zhou, R. Jin, and S. Hoi. Exclusive lasso for multi-task feature selection. In *AISTATS*, 2010.

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