ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/273479604

Multi-label learning with missing labels for image annotation and facial action unit recognition

ARTICLE *in* PATTERN RECOGNITION · FEBRUARY 2015

Impact Factor: 3.1 · DOI: 10.1016/j.patcog.2015.01.022

CITATIONS
3

READS
177

4 AUTHORS:



Baoyuan Wu
King Abdullah University of Science and Te…
7 PUBLICATIONS **18** CITATIONS
SEE PROFILE



Siwei Lyu
University at Albany, The State University o…
62 PUBLICATIONS **1,643** CITATIONS
SEE PROFILE



Bao-Gang Hu
InstiChinese Academy of Sciences
156 PUBLICATIONS **1,908** CITATIONS
SEE PROFILE



Qiang Ji
Tongji Medical University
281 PUBLICATIONS **5,559** CITATIONS
SEE PROFILE

All in-text references underlined in blue are linked to publications on ResearchGate, letting you access and read them immediately.

Available from: Baoyuan Wu
Retrieved on: 17 February 2016

Pattern Recognition ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Multi-label learning with missing labels for image annotation and facial action unit recognition

Baoyuan Wu [a], Siwei Lyu [b], Bao-Gang Hu [a], Qiang Ji [c,*]

[a] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, PR China
[b] Department of Computer Science, University at Albany, SUNY, NY 12222, USA
[c] Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

ARTICLE INFO

Article history:
Received 4 March 2014
Received in revised form
29 November 2014
Accepted 26 January 2015

Keywords:
Multi-label learning
Missing labels
Image annotation
Facial action unit recognition

ABSTRACT

Many problems in computer vision, such as image annotation, can be formulated as multi-label learning problems. It is typically assumed that the complete label assignment for each training image is available. However, this is often not the case in practice, as many training images may only be annotated with a partial set of labels, either due to the intensive effort to obtain the fully labeled training set or the intrinsic ambiguities among the classes. In this work, we propose a method for multi-label learning that explicitly handles missing labels. We train classifiers with the multi-label with missing labels (MLML) learning framework by enforcing the consistency between the predicted labels and the provided labels as well as the local smoothness among the label assignments. Experiments on three benchmark data sets in image annotation and one benchmark data set in facial action unit recognition demonstrate the improved performance of our method in comparison of several state-of-the-art methods.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

In multi-label learning, each example can be associated with many classes simultaneously. The typical examples include image annotation [1] and the recognition of facial action units (AUs) in facial images [2]. More formally, the complete labels of an example $\mathbf{x}$ over m classes, $\{c_1, \ldots, c_m\}$ are represented as a vector $\mathbf{z} \in \{-1, 1\}^m$, where a positive value indicates that the example belongs to the corresponding class, and a negative value indicates the opposite. The prediction from an example $\mathbf{x}$ to its complete label vector, $\mathbf{z}$, is the main task of multi-label learning [3].

The majority of previous multi-label learning methods assume that each example in the training set is associated with a complete label assignment. However, it is difficult to obtain a complete label assignment for each training example. For example, when the size of the candidate classes is large, such as in image annotation (see Fig. 1), it is costly to acquire the complete label vector for even one training image. Another possible reason of this difficulty is the ambiguity among classes, such as in AU recognition. AU is typically labeled by trained experts, which could be a time-consuming process. Furthermore, due to the ambiguity among AUs such as cheek raiser (AU6) vs. eye lid tightener (AU7) (see Fig. 2) as well as the poor image quality, some AUs are difficult to label confidently (the detailed definitions of all AUs can be found in [2]). A more realistic scenario is that each image is only provided with a partial label assignment, while the assignments with other classes are missing (see Fig. 1). To explicitly accommodate the missing labels, for each input image we introduce the definition of an incomplete label vector $\mathbf{y} \in \{-1, 0, 1\}^m$: a nonzero element has the same meaning as in the complete label vector $\mathbf{z}$, while a zero element corresponds to a label with no assigned value for this image, i.e., *missing label*.

The focus of this work is *multi-label learning with missing labels* (MLML) of which the primary objective is to derive a parametric multi-label classifier from input data example $\mathbf{x}$ and its incomplete label vector $\mathbf{y}$. Many previous multi-label learning methods that do handle missing labels usually do not make a clear distinction between missing labels and negative labels, i.e., the missing labels are assigned to the negative value by default [4–6]. However, such a simple treatment is often ungrounded in the practical applications. Consider the image annotation example in Fig. 1, label "home" is not shown in the label list of the image on the left, but we cannot simply conclude that the image should have a negative value for that label, because it has a positive value in the available label "house" that has strong semantic correlation with "home".

We formulate MLML as an inductive learning problem, which consists of three terms, including the consistency between the predicted labels and the provided labels, the smoothness of label

* Corresponding author. Tel.: +1 518 2766313.
E-mail addresses: wubaoyuan1987@gmail.com (B. Wu), lsw@cs.albany.edu (S. Lyu), bghu@nlpr.ia.ac.cn (B.-G. Hu), qji@ecse.rpi.edu (Q. Ji).

http://dx.doi.org/10.1016/j.patcog.2015.01.022
0031-3203/© 2015 Elsevier Ltd. All rights reserved.

Please cite this article as: B. Wu, et al., Multi-label learning with missing labels for image annotation and facial action unit recognition, Pattern Recognition (2015), http://dx.doi.org/10.1016/j.patcog.2015.01.022



Fig. 1. Two examples of image annotation from the ESP Game data [8]. The symbol '√' denotes the positive label, while '×' indicates the negative label. The labels not in the tagging list are missing labels. Take the left image as an example, "house" and "tree" are its positive labels, while "blue" and "forest" are negative labels. All other labels are missing labels. Note that in missing labels, some are actually positive labels, such as "green" and "home", as well as some negative labels, such as "cartoon" and "hat" (see the right image). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)



Fig. 2. An illustration of the ambiguity among AUs [2,9]. Under each facial image, its ground-truth AU labels are annotated (only the AU labels of interest are shown, while other AU labels are ignored): '√' indicates the positive label, while '×' denotes the negative label. Two groups show different ambiguities: (left) the ambiguity between AU6 (cheek raiser) and AU7 (eye lid tightener); (right) the ambiguity between AU23 (lip tightener) and AU24 (lip presser). Due to such ambiguities, some facial images in CK+ database [9] are not labeled with respect to some AU classes.

assignments measured with similarities between examples and between classes, as well as a ℓ_{21} norm over the model parameter to avoid over-fitting. Our algorithm is obtained from the iteratively re-weighted least squares (IRWLS) [7] method. We demonstrate the superior performance of our method on two computer vision problems: one is the annotation of general images, and the other is the recognition of facial action unit (AU) on facial images. The main contributions of our work can be summarized as follows:

(i) we describe a new method to handle missing labels in multi-label learning;
(ii) we present an efficient numerical algorithm to learn the inductive classifier using the MLML framework;
(iii) we formulate image annotation and AU classification as multi-label learning with missing label problems, and show that our method outperforms several state-of-the-art works on three benchmark data sets.

2. Related works

In the literature of multi-label learning, some recent works have studied the problem with missing labels. The semi-supervised multi-label learning (SMSE2) [6] addresses the special case with examples either fully labeled or completely unlabeled. The weak label learning (WELL) method [4] focuses on the case where examples only have a partial set of positive labels available with the rest of the labels unassigned. Both SMSE2 and WELL assign the missing label values to zero by default, which coincides with the numerical value assigned to negative labels. As such, these methods implicitly assume that missing labels are equivalent to negative labels. This assumption is made explicitly in the work of multi-label learning with incomplete class assignment (MLR-GL) [5], where available labels all take positive values, and the missing labels are assigned to negative values, and thus becomes a fully labeled multi-label learning problem. However, as we have argued in Introduction, it is questionable if such an assumption always holds in actual data sets, and whenever it does not, treating missing label indiscriminately as negative labels introduce undesirable bias to the learning problem. The recent work of multi-label learning based on Bayesian compressed sensing (BML-CS) [10] can be used to solve the MLML problem, but it assumes a continuous probability model over the binary labels, and the resulting solution is based on a more costly MCMC algorithm. Moreover, the labels are assumed to be independently distributed in BML-CS, while the proposed model naturally incorporates the correlations between examples and between classes. In another related literature, i.e., matrix completion (MC), some recent works [11–13] have been proposed to handle the MLML problem. Their basic idea is concatenating the label matrix and feature matrix into a unified matrix, based on which the standard matrix completion techniques can be applied to fill in the missing labels. These works also avoid the label bias. However, the low rank assumption in MC implicitly implies class correlations. In contrast, our smoothness assumption explicitly captures such correlations, and we can also replace the smoothness to embed other types of correlations. Besides, the proposed efficient solution only involves matrix multiplication, while the solution to MC exploits the expensive SVD decomposition.

In the literature of image annotation, in addition to the single label methods [14–17], some multi-label learning methods have also been developed, such as [18–24]. However, a common assumption in these methods is that a complete label assignment for each training image should be provided. A few works also consider the case of incomplete labels of training images, such as WELL [4] and MLR-GL [5], which have been mentioned in the first paragraph of this section.

In AU recognition, some existing methods train a binary model with respect to each single class, such as [25–27,9]. The drawback of these methods is the ignorance of the relationships among different AU classes. Some recent models are developed to embed the spatial or temporal relationships among AUs, such as [28–33].

A similar learning problem with MLML, called multi-instance multi-label problem (MIML), has also been applied in image annotation [34–36] and scene classification [37,38]. Both MIML and MLML consider multi-label problems, but there exist two significant differences between them. For clarity, we take the image classification as the example. Firstly, their tasks are different. MIML aims to predict both image-level and region-level labels, while MLML only focuses on image-level label prediction. Secondly, in terms of predicting the image-level labels, MIML can be seen as supervised learning, while MLML can handle the missing labels.

3. Multi-label learning with missing labels

We start with a general setting of the MLML problem. We assume that the data are represented as matrix $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, where each

example is a d-dimensional column vector. Each example can be associated with m different classes $\{c_1, \ldots, c_m\}$, and we are also provided with an available label matrix $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$ with $\mathbf{y}_i \in \{-1, 0, +1\}^{m\times 1}$, whose meanings are as described previously. The definition of Y allows for the inclusion of completely labeled examples, i.e., $\mathbf{y}_i \in \{\pm 1\}^{m\times 1}$, or unlabeled examples, i.e., $\mathbf{y}_i \in \{0\}^{m\times 1}$, or examples with only partial positive or negative labels, i.e., $\mathbf{y}_i \in \{+1, 0\}^{m\times 1}$ or $\{-1, 0\}^{m\times 1}$ respectively.

Our objective is learning a mapping function based on the data X and the provided labels Y, to predict the labels of unlabeled testing examples based on the label consistency and smoothness. Specifically, we want to learn the mapping function $f_{\boldsymbol{\theta}_i}(x_j) = \boldsymbol{\theta}_i^T h(\mathbf{x}_j)$, where $\boldsymbol{\theta}_i \in \mathcal{R}^{d'\times 1}$ denotes the parameter corresponding to the class c_i. $h(\mathbf{x}_j)$ is a general feature function that maps the original d-dimensional data to the d'−dimensional feature. It will be specified in experiments. Given the mapping function, the predicted label Z_{ij} can be determined through a sign function, i.e., $Z_{ij} = \mathrm{sgn}(\boldsymbol{\theta}_i^T h(\mathbf{x}_j))$: $\mathrm{sgn}(a) = 1$ if $a > 0$; $\mathrm{sgn}(a) = -1$ if $a \le 0$.[1]

To achieve this goal, we leverage two assumptions, the *label consistency* and the *label smoothness*. Firstly, we require $Z_{ij} = Y_{ij}$ whenever $Y_{ij} \neq 0$, in other words, the predicted complete labels should be consistent with the available label matrix whenever label i is provided for example $\mathbf{x}_j$. On the other hand, for Z_{ij} where the label information is missing in the available label matrix, i.e., $Y_{ij} = 0$, we will "fill" in their values using two smoothness assumptions:

(i) *Example-level smoothness*: If the features of two images $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar, then their labels represented by the corresponding columns of matrix Z, $Z_{\cdot i}$ and $Z_{\cdot j}$, should also be close.
(ii) *Class-level smoothness*: If two classes c_k and c_l have close semantic meanings, then their instantiations in the overall data set X, represented by the corresponding rows of matrix Z, $Z_{k\cdot}$ and $Z_{l\cdot}$, should also be similar.

We formulate the MLML problem as an optimization problem that maximizes the label smoothness and the label consistency simultaneously.

3.1. Label smoothness

Example-level smoothness: The example-level similarity is computed using a measure defined over each pair of examples and kept in a symmetric positive definite matrix V_Y with $V_Y(i,j) = \mathcal{K}_1(\mathbf{x}_i, \mathbf{x}_j)$, where the similarity kernel function $\mathcal{K}_1(\cdot,\cdot)$ is defined as follows [6]:

$$\mathcal{K}_1(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\varepsilon_i\varepsilon_j}, & i \neq j \\ 0, & i = j, \end{cases} \tag{1}$$

where $\varepsilon_i = \sqrt{\|\mathbf{x}_i - \mathbf{x}_h\|^2}$ and $\mathbf{x}_h$ is the h-th nearest neighbor of $\mathbf{x}_i$.[2] With the example similarity matrix, the example-level smoothness of the complete label matrix Z is measured by

$$\mathcal{S}_Y(Z) = \frac{1}{2}\sum_{k=1}^{m}\sum_{i,j}^{n} V_Y(i,j)\left(\frac{Z_{ki}}{\sqrt{d_Y(i)}} - \frac{Z_{kj}}{\sqrt{d_Y(j)}}\right)^2, \tag{2}$$

where the normalization term $d_Y(i) = \sum_j^n V_Y(i,j)$ makes $\mathcal{S}_Y(Z)$ invariant to the different scaling factors of the elements of V_Y [40]. The above formula can be further simplified by the introduction of the normalized Laplacian matrix $L_Y = I - D_Y^{-1/2} V_Y D_Y^{-1/2}$ with $D_Y = \mathrm{diag}(d_Y(1), \ldots, d_Y(n))$, to

$$\mathcal{S}_Y(Z) = \mathrm{tr}(ZL_YZ^T). \tag{3}$$

Class-level smoothness: The similarities between classes are intrinsic to their semantic meanings. When such similarities are available from other sources, we can use them to construct the class similarity matrix V_C which embeds the semantic similarities between the classes C. However, if such information is absent, we can exploit the co-occurrence relationships among classes, which can be computed based on the available label matrix Y, as did in [6]. Specifically, we define an $m \times m$ nonnegative matrix V_C, as $V_C(i,j) = \mathcal{K}_2(\overline{Y}_{i\cdot}, \overline{Y}_{j\cdot})$, in which the kernel function $\mathcal{K}_2$ is defined as follows [6]:

$$\mathcal{K}_2(\overline{Y}_{i\cdot}, \overline{Y}_{j\cdot}) = \begin{cases} e^{-(1-\langle \overline{Y}_{i\cdot}, \overline{Y}_{j\cdot}\rangle/\|\overline{Y}_{i\cdot}\|\cdot\|\overline{Y}_{j\cdot}\|)/(\varepsilon_i^*\varepsilon_j^*)}, & i \neq j \\ 0, & i = j, \end{cases} \tag{4}$$

where the kernel size $\varepsilon_i^* = \sqrt{1 - \langle \overline{Y}_{i\cdot}, \overline{Y}_{h\cdot}\rangle/(\|\overline{Y}_{i\cdot}\|\cdot\|\overline{Y}_{h\cdot}\|)}$. $\overline{Y}_{i\cdot}$ is a sub-vector of $Y_{i\cdot}$, i.e., $\overline{Y}_{i\cdot} = (Y_{i1}, Y_{i2}, \ldots, Y_{il})$, where l denotes the number of training examples, and $Y_{i\cdot}$ is the i-th row vector of Y. Using the class similarity matrix, we measure the class-level smoothness of Z with

$$\mathcal{S}_C(Z) = \frac{1}{2}\sum_{k=1}^{n}\sum_{i,j}^{m} V_C(i,j)\left(\frac{Z_{ik}}{\sqrt{d_C(i)}} - \frac{Z_{jk}}{\sqrt{d_C(j)}}\right)^2, \tag{5}$$

where $d_C(i) = \sum_j^m V_C(i,j)$ normalizes the factor so that $\mathcal{S}_C(Z)$ is not affected by the different scaling factors of the elements of V_C. Similar to the previous case, we simplify $\mathcal{S}_C(Z)$ by first introducing the normalized Laplacian matrix, i.e., $L_C = I - D_C^{-1/2} V_C D_C^{-1/2}$ with diagonal matrix $D_C = \mathrm{diag}(d_C(1), \ldots, d_C(m))$, and rewriting the above formula as [40]

$$\mathcal{S}_C(Z) = \mathrm{tr}(Z^T L_C Z). \tag{6}$$

3.2. MLML objective function

Combining the two smoothness measures as well as the label consistency term, we formulate the MLML problem as the solution to the following optimization problem:

$$\arg\min_{\Theta} \sum_{i,j}^{m,n} \ell_{\boldsymbol{\theta}_i}(Y_{ij}, h(\mathbf{x}_j)) + \beta\,\mathrm{tr}(ZL_YZ^T) + \gamma\,\mathrm{tr}(Z^T L_C Z), \tag{7}$$

where the two pre-set parameters β and γ control the trade-off between the loss function and two smoothness terms, and can be tuned by cross validation.

We use the hinge loss function $\ell_{\boldsymbol{\theta}_i}(Y_{ij}, h(\mathbf{x}_j)) = \max(0, 1 - Y_{ij}\boldsymbol{\theta}_i^T h(\mathbf{x}_j))$ to enforce the consistency between the predicted label and the provided label Y_{ij}. However, this loss function and the sign function in the definition of Z_{ij} in the smoothness terms are non-differentiable, which makes the numerical optimization of (7) difficult. To alleviate this, we introduce two relaxations to the original problem (7). First the hinge loss function $\ell_{\boldsymbol{\theta}_i}(Y_{ij}, h(\mathbf{x}_j))$ is approximated by a convex differentiable loss function, as follows:

$$\ell_{\boldsymbol{\theta}_i}(Y_{ij}, h(\mathbf{x}_j)) \approx -\log\left(\frac{1}{1 + \exp(-Y_{ij}\boldsymbol{\theta}_i^T h(\mathbf{x}_j))}\right). \tag{8}$$

Second, the sign function $\mathrm{sgn}(\boldsymbol{\theta}_i^T h(\mathbf{x}_j))$ is approximated as

$$\mathrm{sgn}(\boldsymbol{\theta}_i^T h(\mathbf{x}_j)) \approx 2\sigma(\boldsymbol{\theta}_i^T h(\mathbf{x}_j)) - 1 \in [-1, 1], \tag{9}$$

where $\sigma(a) = 1/(1 + \exp(-a))$ denotes the sigmoid function. These two relaxations are illustrated in Fig. 3. Moreover, the $\ell_{2,1}$ norm with respect to $\Theta = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m) \in \mathcal{R}^{d'\times m}$ is used to find sparse representations based on the features. Note that we do not require each individual feature coefficient to be sparse. Each $\boldsymbol{\theta}_i$ may still be dense, but for each data, only a small fraction of Θ will be relevant [41]. As a

[1] Note that there is a little difference with the original definition of the sign function, as we set $\mathrm{sgn}(0) = -1$. Because a possible case is $f_{\boldsymbol{\theta}_i}(\mathbf{x}_j) = 0$, i.e., it fails to predict the label. In this case we just set the predicted label as a negative label, based on the knowledge that most labels are negative in most real applications.

[2] As per the suggestion in [39], we set $h = 7$.

Please cite this article as: B. Wu, et al., Multi-label learning with missing labels for image annotation and facial action unit recognition, Pattern Recognition (2015), http://dx.doi.org/10.1016/j.patcog.2015.01.022

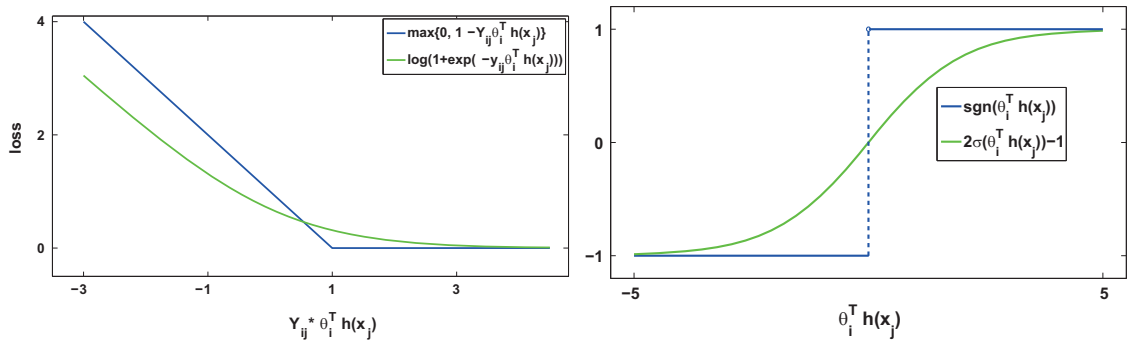


Fig. 3. (left) The hinge loss function and its relaxation; (right) the sign function and its relaxation.

result, our problem becomes learning Θ by minimizing

$$\sum_{i}^{m}\sum_{j}^{n}\log(1+e^{(-Y_{ij}\boldsymbol{\theta}_i^T h(\mathbf{x}_j))})+\beta\mathrm{tr}(\hat{Z}L_Y\hat{Z}^T)+\gamma\mathrm{tr}(\hat{Z}^T L_C\hat{Z})+\eta||\Theta||_{2,1}, \tag{10}$$

where $\hat{Z}=[2\sigma(\Theta^T h(X))-1]\in[-1,1]^{m\times n}$ and $h(X)=(h(\mathbf{x}_1),\ldots,h(\mathbf{x}_n))\in\mathcal{R}^{d'\times n}$. The $\ell_{2,1}$ norm is defined as $||\Theta||_{2,1}=\sum_j^{d+1}\sqrt{\sum_i^m\boldsymbol{\theta}_i^2(j)}$, where $\boldsymbol{\theta}_i^2(j)$ denotes the j-th entry of $\boldsymbol{\theta}_i$.

3.3. Learning

We solve (10) with the iteratively re-weighted least squares (IRWLS) method to handle the non-differentiable $\ell_{2,1}$ term [7,41]. Specifically, we use the fact $|a|=\min_{u>0}\frac{1}{2}(a^2/u+u)$ to replace the $\ell_{2,1}$ term as

$$\arg\min_{\Theta,\mathbf{u}>0}J_\Theta(X)+\frac{\eta}{2}\sum_{j=1}^{d'}\left(\frac{\sum_{i=1}^m\boldsymbol{\theta}_i^2(j)}{u_j}+u_j\right), \tag{11}$$

where $\mathbf{u}=(u_1,\ldots,u_{d'})'\in\mathcal{R}^{d'\times 1}$, and $J_{\boldsymbol{\theta}_i}(X)$ denotes the first three terms in (10). Eq. (11) is solved with coordinate descent, by iteratively minimizing each of the unknown variables, i.e., Θ and $\mathbf{u}$, until convergence.

The optimal solution to $\mathbf{u}$ is thus $u_j^*=\sqrt{\sum_i^m\boldsymbol{\theta}_i^2(j)}, j=1,\ldots,d'$. Given the fixed $\mathbf{u}$, the parameters $\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_m$ can be learned sequentially. Specifically, to optimize $\boldsymbol{\theta}_i$, we seek the solution to the following problem:

$$\arg\min_{\boldsymbol{\theta}_i}J_{\boldsymbol{\theta}_i}(X)+\frac{\eta}{2}\boldsymbol{\theta}_i^T\mathrm{diag}(\mathbf{u})^{-1}\boldsymbol{\theta}_i, \tag{12}$$

which is solved by gradient descent. Specifically, the gradient of the above objective function w.r.t. $\boldsymbol{\theta}_i$ is given by

$$\begin{aligned}\nabla\boldsymbol{\theta}_i=&\eta\cdot\mathrm{diag}(\mathbf{u})^{-1}\boldsymbol{\theta}_i+2\sum_{j=1}^{n}h(\mathbf{x}_j)\cdot\sigma_{ij}(1-\sigma_{ij})\cdot\left(\frac{4\beta}{\sqrt{d_Y(j)}}\right.\\&\cdot\left[\frac{2\sigma_{ij}-1}{\sqrt{d_Y(j)}}\sum_r^n V_Y(j,r)-\sum_r^n\frac{V_Y(j,r)(2\sigma_{ir}-1)}{\sqrt{d_Y(r)}}\right]+\frac{2\gamma}{\sqrt{d_C(i)}}\left[\frac{2\sigma_{ij}-1}{\sqrt{d_C(i)}}\right.\\&\left.\left.\cdot\sum_{r\neq i}^m V_C(i,r)-\sum_{r\neq i}^m\frac{V_C(i,r)(2\sigma_{rj}-1)}{\sqrt{d_C(r)}}\right]\right)\\&-\sum_j^n[1-\sigma(Y_{ij}\boldsymbol{\theta}_i^T h(\mathbf{x}_j))]Y_{ij}h(\mathbf{x}_j),\end{aligned} \tag{13}$$

where σ_{ij} is the shorthand notation for the sigmoid function $\sigma(\boldsymbol{\theta}_i^T h(\mathbf{x}_j))$. Then at the t-th iteration, $\boldsymbol{\theta}_i$ is updated with gradient descent as follows:

$$\boldsymbol{\theta}_i^{(t+1)}=\boldsymbol{\theta}_i^t-\alpha_t\nabla\boldsymbol{\theta}_i^t, \tag{14}$$

where the step size α_t is determined by an optimal step size search based on the Armijo rule [42]. The detailed derivations of the above learning process can be found in 5.

4. Experiments

4.1. Data sets

We evaluate the proposed method on three benchmark data sets in image annotation and AU recognition. Specifically, the used data sets include

(i) *Image annotation*: The three widely used data sets, ESP Game [8], MIR Flickr [43] and NUS-WIDE-Lite [44], are adopted in our experiments. With a similar setting as [5], we delete the images of few positive labels and some rare classes. Specifically, in ESP Game, we delete the images with less than 5 positive labels and the classes that exist in less than 300 images. Then ESP Game consists of 10,457 images with 54 classes, including 9418 training and 1039 testing images. In MIR Flickr, images with less than 3 positive labels and classes that exist in less than 300 images are deleted. Then 10,199 images with 47 classes are remained, including 5137 training images and 5062 testing images. In NUS-WIDE-Lite, images with less than 7 positive labels and classes that exist in less than 150 images are deleted, then 2456 images with 23 classes are remained. The positive label proportions in the whole ground-truth label matrix are 11.88%, 5.30% and 29.4% in ESP Game, MIR Flickr and NUS-WIDE-Lite respectively. In ESP Game and MIR Flickr, each image is represented by the RGB features in a 4096-dimensional vector [45].[3] In NUS-WIDE-Lite, each image is represented by a 265-dimensional vector, which consists of three types of features: 64-dimensional color histogram; 73-dimensional edge direction histogram; 128-dimensional wavelet texture.[4] For the above three data sets, the Euclidean distance is used as the distance between images. The original dimension is reduced by PCA, in order to reduce the computational cost.

(ii) *AU recognition*: The benchmark data set in AU recognition, the Extended Cohn-Kanade (CK+) database [9] is adopted. 327 facial images with 16 most frequent AU classes (including AU 1, 2, 4, 5, 6, 7, 9, 12, 14, 15, 17, 20, 23, 24, 25, and 27) are chosen from the whole database. The positive label proportion in the whole ground-truth label matrix is 24.94%. Each facial image is described by a 201-dimensional column vector, which is

[3] The features are downloaded from 'http://lear.inrialpes.fr/people/guillaumin/data.php'.

[4] The features are download from 'http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm'.

Table 1
Data statistics

Data set	Example	Class	Feature	Avg. posi.-class/example	Avg. posi.-example/class	Posi. label proportion (%)
ESP Game [8]	10,457	54	4096	6.41	1242.2	11.88
MIR Flickr [43]	10,199	47	4096	1.22	540.39	5.30
NUS-WIDE-Lite [44]	2456	23	265	6.76	722.13	29.40
AU [9]	327	16	201	3.99	81.5	24.94

cascaded by four types of features: 102-dimensional vector of the locations of 51 facial feature points; 40-dimensional texture features; 30-dimensional appearance features; 29-dimensional shape features.

The basic statistics of above data sets are summarized in Table 1. For NUS-WIDE-Lite and AU data, we randomly partition the whole data set to 5 uniform folds, and 4 folds are used as the training set while the remaining one is used as the testing set. Consequently we obtain 5 results. We further repeat this process 5 runs to obtain different partitions. Then we will obtain 25 results in total. Finally the mean and the standard deviation of the evaluation scores (the evaluation metrics will be specified later) of the 25 results are computed as the outputs. For all data sets, to simulate different scenarios with missing labels, we create training sets with varying portions of available labels, from 20% (i.e., 80% missing labels) to 100% (i.e., no missing labels). In each case, the missing labels are randomly chosen and removed from the ground-truth label matrix of the training data. We perform the experiments 5 runs to obtain different missing labels. In all cases, the experimental results of the testing sets are reported, which are summarized as the mean and standard deviation.

4.2. Experiment settings

Initializations: The initialization of the parameters $\boldsymbol{\Theta}$ in the proposed method is performed as follows: for each single class, a logistic regression with the ℓ_1 norm is trained (implemented by the built-in function "*lassoglm*" in Matlab software), then the learned $\boldsymbol{\theta}_i$ is used as the initialization of our model parameter. The feature function $h(\mathbf{x}_j)$ is simply specified as $[1; x_j]$ in our experiments, then $d' = d+1$. Parameters β, γ and η in Eq. (10) are tuned by cross validation with the range $[10^{-2}, 10^{2}]$, $[10^{-2}, 10^{2}]$ and $[10^{-2}, 10^{3}]$ respectively.

Evaluation metrics: As the predicted continuous labels can be seen as a ranking list of each sample, we adopt two widely used performance metrics for multi-label ranking, namely, average precision (AP) [46] and area under ROC curve (AUC) [5]. Specifically, AP is calculated as follows [46]:

$$\mathrm{AP} = \frac{1}{n}\sum_{i}^{n}\frac{1}{|S^i|}\sum_{s_r \in S^i}\frac{|\{s_t \in S^i \,|\, \mathrm{rank}(\mathbf{x}_i, s_t) < \mathrm{rank}(\mathbf{x}_i, s_r)\}|}{\mathrm{rank}(\mathbf{x}_i, s_r)}, \tag{15}$$

where S^i denotes the set of ground-truth positive classes of sample $\mathbf{x}_i$, and $\mathrm{rank}(\mathbf{x}_i, s_t)$ represents the rank order of class s_t in the label ranking list of $\mathbf{x}_i$. AUC [5] measures the total area under ROC curve (true positive rate vs. false positive rate) of all classes. Specifically, given the label ranking list for each sample, the first k classes are treated as positive, while others are treated as negative. Through varying k from 1 to m (the number of classes), the ROC curve of all classes is gained. Higher values of both metrics represent better performances. Note that both metrics are the overall evaluations of the label ranking list, hence the score corresponding to each class is not calculated.

Comparisons: We compare the proposed method with several previous works on multi-label learning with missing labels and image annotation, including SMSE2 [6], MLR-GL [5] and MC-Pos [12]. We implement SMSE2 in Matlab and adopt the publicly available code MLR-GL. The code of MC-Pos is provided by its authors. We make our best effort to adjust the parameters in these methods as suggested in the original papers. We also include two baselines: one is learning the logistic regression with ℓ_1 norm (it is also the initialization method of our algorithm, as mentioned above), and the other is learning a binary SVM[5] classifier, for each single class independently. They both only use the labeled examples as the training set in each individual class, while the images with missing labels are ignored. Besides, we evaluate the influences of the example-level and class-level smoothness in the objective function (7), by setting $\beta = 0$ and $\gamma = 0$ respectively. They are denoted as $\mathrm{MLML}_{\beta=0}$ and $\mathrm{MLML}_{\gamma=0}$.

4.3. Results

Image annotation: The results of the image annotation task are shown in Tables 2–4. On ESP Game data, the proposed method MLML gives better results than other methods in most cases. On MIR Flickr data, the performance of MLML is better results than other methods in the cases of high label proportions, while is worse when the label proportion is low. Then possible reasons include when the label proportion is low, the computed class correlations based on concurrency are likely to be far from the ground-truth class correlations. Consequently the performance of MLML may be harmed by such a poor class correlation. Besides, compared with MC-Pos, the proposed inductive model may be more sensitive to the unbalanced label distribution problem. On NUS-WIDE-Lite data, the proposed method MLML also performs better than other methods in most cases. The comparisons between MLML and $\mathrm{MLML}_{\beta=0}$, $\mathrm{MLML}_{\gamma=0}$ are also presented. On ESP Game, $\mathrm{MLML}_{\beta=0}$ performs better than MLML in some cases. It demonstrates that the example-level smoothness does not provide the positive information to help MLML. The possible reason is that the Euclidean distance in $\mathcal{K}_1$ is not the best choice to reveal the example-level correlations on this data. The performance is expected to be further improved by exploiting more sophisticated distances. On MIR Flickr, MLML does not show consistent superiority to $\mathrm{MLML}_{\beta=0}$ and $\mathrm{MLML}_{\gamma=0}$. This demonstrates that the influences of example-level and class-level smoothness depend on label distribution and the provided label proportion. Moreover, the overall performance of all methods on these two data sets is not very high. It may be due to the simple image features and the image distance used in our experiments. However, it is fair enough to compare the performance of different methods. The comparisons on NUS-WIDE-Lite show that $\mathrm{MLML}_{\gamma=0}$ performs better than MLML in the cases of low provided label proportions. The possible reason is when many missing labels exist, the computed kernel $\mathcal{K}_2$ is far from the ground-truth semantic dependencies among classes. As such, it may harm the model performance. Furthermore, the symbol "–" in Tables 2, 3 and 5 indicates that the method does not work or is inapplicable in the corresponding case. Specifically, SMSE is developed for semi-supervised learning. Some annotation results are shown in Table 6.

[5] The SVM classifiers are trained with the LIBSVM package [47] with parameters estimated by cross-validation.

Please cite this article as: B. Wu, et al., Multi-label learning with missing labels for image annotation and facial action unit recognition, Pattern Recognition (2015), http://dx.doi.org/10.1016/j.patcog.2015.01.022

Table 2
AP and AUC results on ESP Game data (mean(std)%), see texts for details. The best result in each column is highlighted in bold.

Methods	AP				AUC			
	20%	40%	80%	100%	20%	40%	80%	100%
Logistic-ℓ_1	29.28(0.56)	28.52(0.27)	27.27(1.12)	27.83(0.00)	70.93(0.29)	70.21(1.21)	68.63(0.53)	70.19(0.00)
SVM [47]	19.74(1.80)	18.00 (0.20)	18.77(0.15)	19.58(0.00)	60.77(0.90)	56.57(1.01)	53.51(2.18)	54.56(0.00)
SMSE2 [6]	–	–	–	23.50(0.00)	–	–	–	67.16(0.00)
MLR-GL [5]	27.18(1.14)	26.53(0.43)	26.09(0.56)	26.24(0.00)	69.84(0.10)	69.79(0.46)	69.59(0.16)	69.28(0.00)
MC-Pos [12]	28.57(0.07)	25.60(0.22)	24.53(0.07)	24.60(0.00)	68.70(0.17)	66.56(0.01)	62.21(0.05)	63.14(0.00)
$\text{MLML}_{\beta=0}$	29.93(0.87)	29.42(0.22)	29.46(0.67)	29.49(0.00)	71.04(0.45)	**71.16(0.83)**	70.96(0.25)	**71.15(0.00)**
$\text{MLML}_{\gamma=0}$	29.89(0.11)	29.26(0.55)	29.12(0.22)	29.38(0.00)	**71.21(0.69)**	70.95(0.44)	70.87(0.33)	70.98(0.00)
MLML	**29.97(0.87)**	**29.75(0.27)**	**29.74(0.38)**	**29.67(0.00)**	71.14(0.61)	70.94(0.28)	**71.05(0.28)**	71.05(0.00)

Table 3
AP and AUC results on MIR Flickr data (mean(std)%), see texts for details. The best result in each column is highlighted in bold.

Methods	AP				AUC			
	20%	40%	80%	100%	20%	40%	80%	100%
Logistic-ℓ_1	23.94(0.48)	25.42(1.32)	24.42(0.55)	24.43(0.00)	59.73(0.42)	60.31(0.25)	59.81(0.36)	60.38(0.00)
SVM [47]	18.25(0.71)	19.96(1.13)	18.30(0.72)	18.92(0.00)	53.50(4.62)	55.05(1.38)	52.27(2.38)	54.85(0.00)
SMSE2 [6]	–	–	–	21.00(0.00)	–	–	–	58.29(0.00)
MLR-GL [5]	**26.48(0.12)**	26.33(0.25)	25.81(0.08)	25.72(0.00)	**60.17(0.22)**	59.64(0.44)	58.59(0.01)	58.31(0.00)
MC-Pos [12]	26.43(0.08)	**26.56(0.21)**	22.45(0.07)	22.37(0.00)	59.61(0.01)	60.12(0.44)	59.89(0.41)	59.64(0.00)
$\text{MLML}_{\beta=0}$	24.42 (0.46)	26.46 (0.46)	25.61(0.18)	26.51(0.00)	59.98(0.34)	60.42(0.22)	**60.20(0.49)**	60.41(0.00)
$\text{MLML}_{\gamma=0}$	24.45(0.82)	26.42(0.79)	24.22(0.15)	24.88(0.00)	60.13(0.77)	60.66(0.13)	59.82(0.35)	**60.46(0.00)**
MLML	24.46(0.37)	26.41(0.29)	**25.83(0.17)**	**26.56(0.00)**	60.10(0.45)	**60.84(0.26)**	60.10(0.25)	60.45(0.00)

Table 4
AP and AUC results on NUS-WIDE-Lite data (mean(std)%), see texts for details. The best result in each column is highlighted in bold.

Methods	AP				AUC			
	20%	40%	80%	100%	20%	40%	80%	100%
Logistic-ℓ_1	82.77(0.11)	83.67(0.07)	84.17(0.04)	84.34(0.03)	88.59(0.27)	89.56(0.12)	90.02(0.05)	90.19(0.05)
SVM [47]	82.97(0.09)	83.54(0.05)	84.10(0.07)	84.27(0.08)	88.87(0.08)	89.48(0.03)	89.95(0.05)	90.11(0.03)
SMSE2 [6]	–	–	–	84.40(0.06)	–	–	–	90.16(0.05)
MLR-GL [5]	80.50(0.14)	82.16(0.09)	83.35(0.03)	83.64(0.04)	87.06(0.09)	88.41(0.05)	89.32(0.04)	89.57(0.03)
MC-Pos [12]	81.48(0.05)	81.35(0.08)	81.53(0.02)	81.51(0.03)	87.46(0.16)	87.27(0.09)	87.44(0.02)	87.49(0.03)
$\text{MLML}_{\beta=0}$	83.08(0.05)	83.86(0.09)	84.41(0.04)	84.48(0.02)	89.11(0.06)	89.84(0.06)	90.27(0.03)	90.31(0.01)
$\text{MLML}_{\gamma=0}$	**83.10(0.07)**	**83.95(0.01)**	84.47(0.05)	84.48(0.01)	**89.18(0.13)**	89.87(0.11)	90.29(0.08)	90.28(0.01)
MLML	83.08(0.08)	83.91(0.01)	**84.49(0.02)**	**84.56(0.07)**	89.00(0.11)	**89.91(0.02)**	**90.32(0.02)**	**90.34(0.08)**

Table 5
AP and AUC results on AU data (mean(std)%), see texts for details. The best result in each column is highlighted in bold.

Methods	AP				AUC			
	20%	40%	80%	100%	20%	40%	80%	100%
Logistic-ℓ_1	78.11(0.59)	83.75(0.03)	87.64(0.12)	88.26(0.33)	86.80(0.08)	90.22(0.42)	93.59(0.12)	94.06(0.13)
SVM [47]	**81.05(0.94)**	85.30(0.85)	88.18(0.75)	89.03(0.46)	89.49 (0.51)	92.06 (0.45)	92.53 (0.26)	94.03 (0.16)
SMSE2 [6]	–	–	–	85.74(0.06)	–	–	–	92.41(0.12)
MLR-GL [5]	79.14(1.60)	83.33(0.35)	86.47(0.39)	87.43(0.52)	88.90(0.69)	91.50 (0.19)	93.12 (0.21)	93.42 (0.26)
MC-Pos [12]	78.28(0.31)	83.37(0.33)	87.90(0.26)	90.04(0.20)	87.66(0.19)	90.86(0.38)	93.78(0.23)	94.57(0.16)
$\text{MLML}_{\beta=0}$	79.93(0.27)	84.72(0.52)	89.18(0.10)	89.68(0.16)	88.83(0.82)	92.02(0.31)	**94.47(0.14)**	94.77(0.09)
$\text{MLML}_{\gamma=0}$	81.01(0.39)	85.07(1.06)	87.40(0.49)	89.83(0.03)	**89.59(0.11)**	92.11(0.68)	93.51(0.33)	**94.89(0.22)**
MLML	80.98(0.91)	**85.49(0.48)**	**89.27(0.53)**	**90.17(0.49)**	89.15(0.18)	92.19(0.99)	94.09(0.26)	**94.89(0.02)**

AU recognition: The results on the AU data are presented in Table 5. The proposed method MLML outperforms than other methods in most cases. When the label proportion is 20%, the result of MLML is slightly lower than the result of SVM. The possible reason is the class-level smoothness (see Eq. (4)). When the label proportion is low, the computed similarities between classes may be inaccurate. This point is also proved by the comparison between MLML and $\text{MLML}_{\gamma=0}$. It suggests that exploiting a more robust class-

Please cite this article as: B. Wu, et al., Multi-label learning with missing labels for image annotation and facial action unit recognition, Pattern Recognition (2015), http://dx.doi.org/10.1016/j.patcog.2015.01.022

Table 6
Examples of testing images from the ESP Game and MIR Flickr data with the predicted positive labels of different methods. The correct labels are highlighted in bold.

Images	Ground-truth	Logistic-ℓ_1	SVM [47]	SMSE2 [6]	MLR-GL [5]	MC-Pos [12]	MLML
	**black, blue, grass, hair, hat, man, people**	**black, blue**, white, tree, sky, **man, hair**	**blue**, water, painting, tree, gray, sky, white	**blue, man**, red, green, white, woman, tree	**blue, man**, red, woman, green, **people, hair**	water, sky, **blue**, white, sea, tree, ocean	**man**, white, **blue, hair, people, black**, woman
	**explore, sky, nikon, blue, clouds**	green, **blue**, red, **explore, sky**, yellow	green, **blue**, flower, white, interestingness, beach	**explore, sky, blue, nikon, clouds**, nature	**explore**, green, **blue, sky**, nature, **nikon**	**explore, sky, blue**, nature, **clouds**	**explore, sky, blue, clouds**, nature, **nikon**

Table 7
Examples of testing images from the AU data with the predicted positive labels of different methods. The correct labels are highlighted in bold.

Images	Ground-truth	Logistic-ℓ_1	SVM [47]	SMSE2 [6]	MLR-GL [5]	MC-Pos [12]	MLML
	**AU14**	AU4	AU25	AU1	AU1	AU4	**AU14**
	**AU1, AU4, AU15, AU17**	**AU4, AU17, AU1, AU15**	**AU1**, AU20, **AU4**, AU25	**AU1**, AU25, AU2, AU5	**AU1, AU4, AU17, AU15**	**AU4**, AU7, **AU1**, AU5	**AU4, AU17, AU1, AU15**
	**AU1, AU2, AU4, AU5, AU14, AU20, AU25**	**AU25, AU5, AU1, AU20, AU4, AU2, AU14**	**AU20, AU25, AU1, AU5, AU2**, AU6, AU12	**AU25, AU4, AU1, AU20, AU5**, AU7, AU6	**AU25, AU1, AU20, AU4, AU5**, AU12, **AU14**	**AU25, AU5, AU4, AU20, AU1**, AU12, AU6	**AU25, AU5, AU20, AU1, AU4, AU2, AU14**

level smoothness independent of the provided labels may further boost the performance of MLML. This will be explored in our future work. The predicted labels on some facial images are shown in Table 7.

4.4. Top-5 accuracy of each class

In above evaluations, the results are calculated based on the label ranking list of all classes. To evaluate the performance of each class, in the following we show the results evaluated by the top-5 accuracy, in the case of 100% label proportion. Specifically, for the label ranking list of each testing example, we assign the first five classes as positive classes, while all other classes as negative. Hence we obtain a discrete label matrix. Based on this matrix, we compute the average top-5 accuracy, which equals to the fraction of the correct prediction labels in ground-truth label matrix. The corresponding results of different algorithms are summarized in Table 8.

The top-5 accuracy of each class is also shown in Fig. 4. Note that all compared algorithms, including the proposed MLML algorithm, show poor performance on ESP Game and MIR Flickr data. We believe that the main reason is the extremely unbalanced label distribution, i.e., the positive labels are significantly less than the negative labels. As shown in Table 1, the positive label percentage are only 11.88% and 5.30% in ESP Game and MIR Flickr respectively. Actually, the unbalanced label distribution is a typical problem in multi-label learning. This motivates us to design more robust model, such as the weighted model. This will be explored in our future work.

4.5. Computational complexity and convergence

The computational complexity of the proposed algorithm and other compared algorithms are summarized in Table 9. The main computational cost of our algorithm is the computation of $\nabla\boldsymbol{\theta_i}$ (see Eq. (13)), which only involves matrix multiplication. Its computational complexity is $O(n(m+n+d))$. Moreover, by setting V_Y and V_C as sparse matrix, the complexity can be further reduced significantly. As $\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_m$ are learned sequentially, the overall complexity of the proposed algorithm is $O(mn(m+n+d))$. It is comparable with MLR-GL, and is much faster than other algorithms. However, when the number of candidate classes m is large, the complexity of MLML will be high. This is why we only choose a part of candidate classes in above experiments. To alleviate this problem, the label space reduction can be exploited to be combined with our current model. This will be explored in our future work.

Table 8
Comparisons on the average top-5 accuracy (%) of different algorithms. The best result in each column is highlighted in bold.

Methods	Data sets			
	ESP Game	MIR Flickr	NUS-WIDE-Lite	AU
Logistic-ℓ_1	19.69	7.48	81.82	70.76
SVM [47]	13.13	5.64	81.93	70.77
SMSE2 [6]	15.48	6.20	80.76	63.25
MLR-GL [5]	19.87	6.80	**82.16**	68.45
MC-Pos [12]	20.45	7.07	79.06	60.42
$\text{MLML}_{\beta=0}$	21.58	**7.78**	82.01	71.46
$\text{MLML}_{\gamma=0}$	21.46	7.48	82.10	71.25
MLML	**21.62**	**7.78**	82.08	**71.49**

The convergence curves of the proposed algorithm on all data sets are shown in Fig. 5. Considering the number of classes in each data set, the proposed algorithm often converges in a small number of iterations. It guarantees the computational efficiency of our algorithm.

4.6. Discussions

According to the above results, we find that in different proportions of missing labels, the performances of MLML seem to be inconsistent. Indeed in Section 1 we have demonstrated that

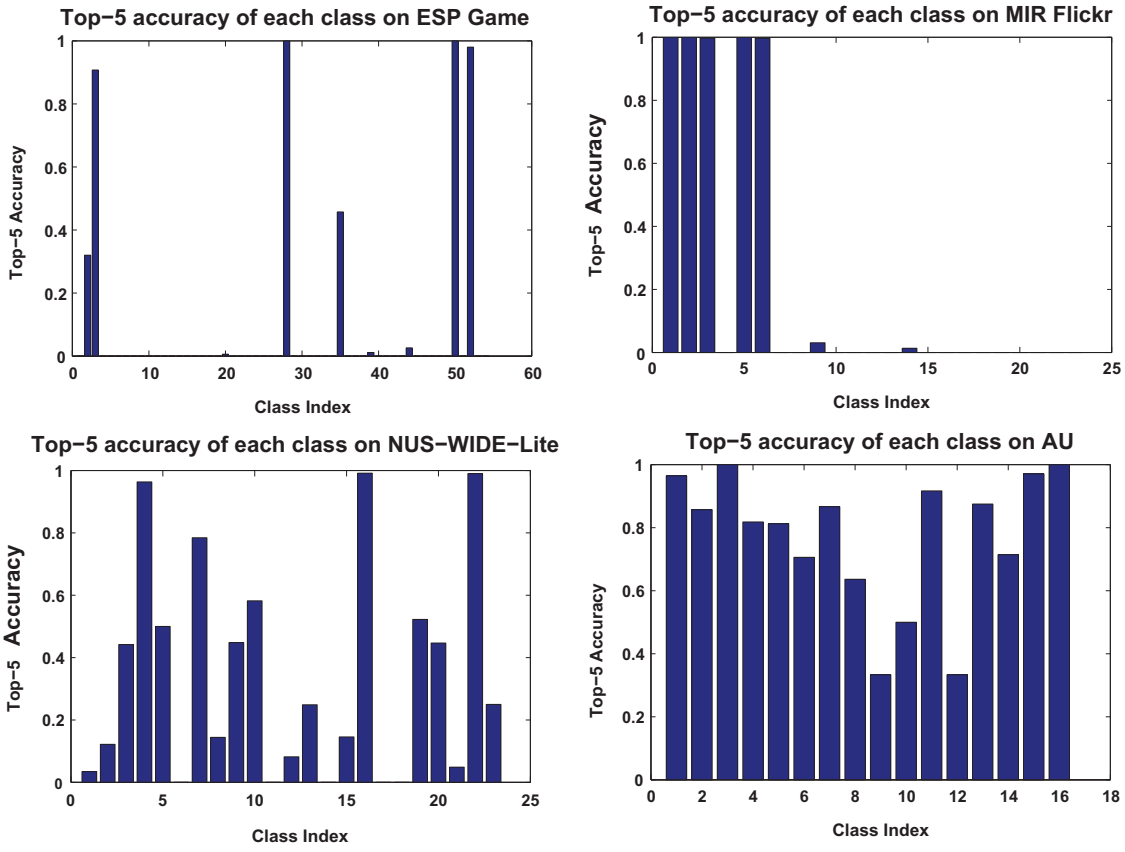


Fig. 4. Top-5 accuracy on of the proposed algorithm on different data sets.

Table 9
Computational complexities of different algorithms.

Algorithm	SMSE2 [6]	MLR-GL [5]	MC-Pos [12]	MLML
Complexity	$O(n^3)$	$O(mn^2)$	$O(mn^2+n^3)$	$O(mn(m+n+d))$

MLML method can avoid the label bias (i.e., treating missing labels as negative labels directly), which exists in some compared methods, such as SMSE2, WELL and MLR-GL. However, this advantage cannot always guarantee that when the amount of missing labels increases, the performance of MLML becomes better than other methods. We believe that the reason is the performance of MLML is not only influenced by the amount of missing labels, but is also influenced by other factors, including the percentage of positive labels in the ground-truth label matrix, as well as the initial parameters from binary logistic regression. Specifically, firstly when many missing labels exist, the computed class-level dependency based on the provided label vector (i.e., $\mathcal{K}_2$, see Eq. (4)) may be far from the true semantic dependencies among classes. Secondly, when many missing labels exist, the binary logistic regression, which only exploits the labeled examples in each class, may not provide good initial parameters for MLML. These two points may harm the performance of MLML. Thirdly, if the positive label percentage is very low, such as the 8.83% in MIR Flickr, the label bias may not significantly harm the performance of other methods. Because treating missing as negative labels will bring in many correct negative labels, while the number of wrong labels (treating the ground-truth positive labels as negative labels) is small. As a result, the benefit from the correct labels may compensate the harm from the wrong labels.

Moreover, we also observe that in many cases the performance superiority of MLML is not very significant over the compared methods. We believe that there are three possible reasons. Firstly, MLML can be seen as an extension of linear logistic regression model. The linear model may limit the performance of MLML. It inspires us to make kernelization of MLML, which is expected to further enhance the performance. Secondly, the class-level dependency $\mathcal{K}_2$ depends on the provided labels of the specific data. It inspires us to explore and exploit some other types of dependencies independent of the provided labels, which is expected to make the performance of MLML more robust. Last, the example-level dependency $\mathcal{K}_1$ depends on the hand-crafted image features. In terms of image-based problems, visually similar images may not always lead to the closeness of labels. This semantic gap may harm the model performance. It inspires us to extract or learn more informative features to describe the semantic meaning of the image, based on saliency detection [48] or deep learning [49].

5. Conclusions and future work

In this paper, we introduce a new formulation and solution to multi-label learning with missing labels, and demonstrate its effectiveness on two computer vision applications, image annotation and AU recognition. Different from the existing methods, our model explicitly takes account of missing labels and systematically incorporates the label consistency and smoothness into the solution. Experiments on three benchmark data sets have verified the efficacy of the proposed method.

As future work, there are several directions we would like to further study. First, in our current work we only adopt the simple concurrency in the class-level smoothness and the simple Euclidean distance in example-level smoothness. Other types of class correlations and image distances have been explored in many existing works in the literature of multi-label learning and image

Please cite this article as: B. Wu, et al., Multi-label learning with missing labels for image annotation and facial action unit recognition, Pattern Recognition (2015), http://dx.doi.org/10.1016/j.patcog.2015.01.022

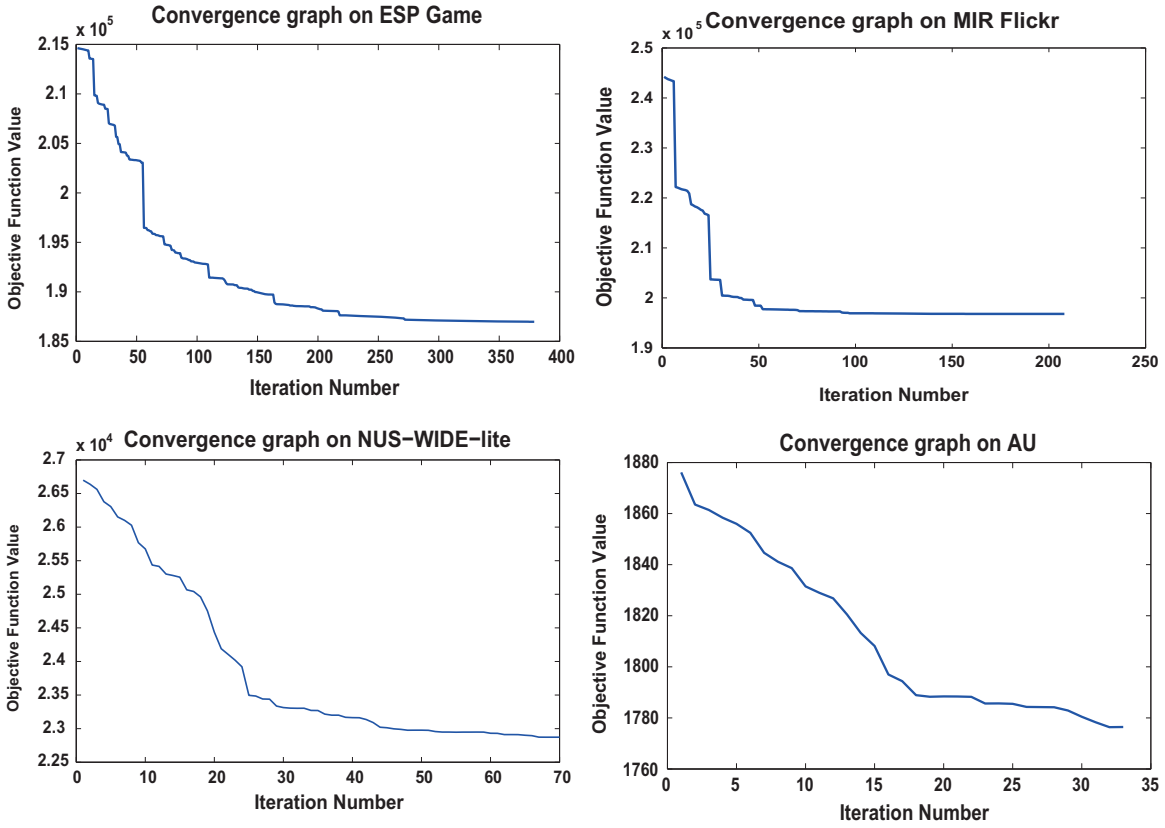


Fig. 5. Illustrations of the convergence rate of the proposed algorithm. One iteration number corresponds to one update step of $\boldsymbol{\theta}_i$ (see Eq. (19)).

annotation respectively [50,51]. Combined with these works, the performance of our model is expected to be further improved. Second, we only use the linear logistic regression in our model. However, our model can be kernelized by setting different feature functions $h(\mathbf{x}_j)$. The kernelized model is expected to handle the data with more complex distributions. Third, as discussed in Section 4, the proposed MLML model still suffers from two problems: one is the scalability to the large scale class space, and the other is the unbalanced label distribution. They could be alleviated by label space reduction and the weighted model respectively. Fourth, sometimes the soft labels (labels with uncertainty) occur in particular scenarios, such as the AU intensity . Handling and predicting such soft labels can be seen as regression problems, such as [52]. As the discrete label Z_{ij} is approximated by a continuous curve (see Eq. (9) and Fig. 3) in our learning process, our method can directly handle the provided soft labels and also output soft labels. As such it is a potential direction to extend our method for the multi-label regression problems. Last, we will also explore other applications of MLML, such as image denoising, image segmentation and scene classification.

Conflict of interest

None declared.

Acknowledgments

The work was mostly completed when the first author was a visiting student at Rensselaer Polytechnic Institute (RPI), supported by a scholarship from China Scholarship Council (CSC). We thank CSC and RPI for their support. Qiang Ji is supported in part by a grant from the US National Science Foundation (NSF, No. 1145152). Siwei Lyu is supported in part by the US National Science Foundation Research Grant (CCF-1319800) and the National Science Foundation Early Faculty Career Development (CAREER) Award (IIS-0953373). Bao-Gang Hu and Baoyuan Wu are supported in part by the National Natural Science Foundation of China (NSFC, No. 61273196).

Appendix A: optimization of the MLML objective function

For convenience, we firstly repeat the objective function (see Eq. (11) in Section 3.3 of the main paper) as follows:

$$\arg\min_{\Theta,\mathbf{u}>0} J_{\Theta}(X)+\frac{\eta}{2}\sum_{j=1}^{d}\left(\frac{\sum_{i=1}^{m}\boldsymbol{\theta}_i^2(j)}{u_j}+u_j\right). \tag{16}$$

In more detail, the above objective function can be expanded as follows (in the following, we will use σ_{ij} as the shorthand notation for $\sigma(\boldsymbol{\theta}_i^T h(\mathbf{x}_j))$):

$$\begin{aligned}\arg\min_{\Theta,\mathbf{u}} \quad & \beta\sum_{i}^{m}\sum_{j,r}^{n}V_Y(j,r)\left(\frac{2\sigma_{ij}-1}{\sqrt{d_Y(j)}}-\frac{2\sigma_{ir}-1}{\sqrt{d_Y(r)}}\right)^2\\ &+\gamma\sum_{i}^{m}\sum_{r\neq i}^{m}\sum_{j}^{n}V_C(i,r)\\ &\times\left(\frac{2\sigma_{ij}-1}{\sqrt{d_C(i)}}-\frac{2\sigma_{rj}-1}{\sqrt{d_C(r)}}\right)^2-\sum_{i}^{m}\sum_{j}^{n}\log\ \sigma(Y_{ij}\boldsymbol{\theta}_i^T h(\mathbf{x}_j))\\ &+\frac{\eta}{2}\sum_{j=1}^{d'}\left(\frac{\sum_{i=1}^{m}\boldsymbol{\theta}_i^2(j)}{u_j}+u_j\right),\end{aligned} \tag{17}$$

where $\boldsymbol{\theta}_i(j)$ denotes the j-th entry of $\boldsymbol{\theta}_i$. This problem is solved with coordinate descent, with iteratively updating Θ and $\mathbf{u}$, until convergence.

.1. Given Θ, learning $\mathbf{u}$

The optimal solution to $\mathbf{u}$ can be easily gained by setting the gradient of (17) w.r.t. $\mathbf{u}$ to be zero, as follows:

$$-\frac{\sum_{i=1}^{m}\boldsymbol{\theta}_i^2(j)}{u_j^2}+1=0,\quad \text{with } u_j>0\Longrightarrow u_j^*=\sqrt{\sum_{i=1}^{m}\boldsymbol{\theta}_i^2(j)}. \tag{18}$$

.2. Given $\mathbf{u}$, learning Θ

Given the fixed $\mathbf{u}$, the last term of Eq. (17) becomes

$$\sum_{j=1}^{d'}\left(\frac{\sum_{i=1}^{m}\boldsymbol{\theta}_i^2(j)}{u_j}+u_j\right)=\sum_{i=1}^{m}\sum_{j=1}^{d'}\frac{\boldsymbol{\theta}_i^2(j)}{u_j}+\sum_{j=1}^{d'}u_j$$
$$=\sum_{i=1}^{m}\boldsymbol{\theta}_i^T\operatorname{diag}(\mathbf{u})^{-1}\boldsymbol{\theta}_i+\sum_{j=1}^{d'}u_j.$$

Obviously this part becomes a weighted least squares. Furthermore, when all other parameters $\Theta/\boldsymbol{\theta}_i$ and $\mathbf{u}$ are fixed, the objective function with respect to $\boldsymbol{\theta}_i$ becomes

$$\arg\min_{\boldsymbol{\theta}_i}\ \beta\sum_{j,r}^{n}V_Y(j,r)\left(\frac{2\sigma_{ij}-1}{\sqrt{d_Y(j)}}-\frac{2\sigma_{ir}-1}{\sqrt{d_Y(r)}}\right)^2+\gamma\sum_{r\neq i}^{m}\sum_{j}^{n}V_C(i,r)$$
$$\cdot\left(\frac{2\sigma_{ij}-1}{\sqrt{d_C(i)}}-\frac{2\sigma_{rj}-1}{\sqrt{d_C(r)}}\right)^2-\sum_{j}^{n}\log\,\sigma(Y_{ij}\boldsymbol{\theta}_i^Th(\mathbf{x}_j))+\frac{\eta}{2}\boldsymbol{\theta}_i^T\operatorname{diag}(\mathbf{u})^{-1}\boldsymbol{\theta}_i. \tag{19}$$

Next, denote the four terms in (19) as A, B, C, and D. Then their respective gradient w.r.t. $\boldsymbol{\theta}_i$ are as follows:

$$\frac{\partial A}{\partial\boldsymbol{\theta}_i}=2\beta\sum_{j,r}^{n}V_Y(j,r)\left(\frac{2\sigma_{ij}-1}{\sqrt{d_Y(j)}}-\frac{2\sigma_{ir}-1}{\sqrt{d_Y(r)}}\right)\left(\frac{2\sigma_{ij}(1-\sigma_{ij})h(\mathbf{x}_j)}{\sqrt{d_Y(j)}}-\frac{2\sigma_{ir}(1-\sigma_{ir})h(\mathbf{x}_r)}{\sqrt{d_Y(r)}}\right)$$
$$=4\beta\sum_{j,r}^{n}V_Y(j,r)\cdot\left(\frac{(2\sigma_{ij}-1)\sigma_{ij}(1-\sigma_{ij})h(\mathbf{x}_j)}{d_Y(j)}\right.$$
$$+\frac{(2\sigma_{ir}-1)\sigma_{ir}(1-\sigma_{ir})h(\mathbf{x}_r)}{d_Y(r)}-\frac{(2\sigma_{ij}-1)\sigma_{ir}(1-\sigma_{ir})h(\mathbf{x}_r)}{\sqrt{d_Y(j)d_Y(r)}}$$
$$\left.-\frac{(2\sigma_{ir}-1)\sigma_{ij}(1-\sigma_{ij})h(\mathbf{x}_j)}{\sqrt{d_Y(r)d_Y(j)}}\right)=8\beta\sum_{j}^{n}h(\mathbf{x}_j)\sigma_{ij}(1-\sigma_{ij})$$
$$\cdot\left(\frac{(2\sigma_{ij}-1)\sum_{r}^{n}V_Y(j,r)}{d_Y(j)}-\frac{1}{\sqrt{d_Y(j)}}\sum_{r}^{n}\frac{(2\sigma_{ir}-1)V_Y(j,r)}{\sqrt{d_Y(r)}}\right),$$

$$\frac{\partial B}{\partial\boldsymbol{\theta}_i}=2\gamma\sum_{r\neq i}^{m}\sum_{j}^{n}V_C(i,r)\left(\frac{2\sigma_{ij}-1}{\sqrt{d_C(i)}}-\frac{2\sigma_{rj}-1}{\sqrt{d_C(r)}}\right)\frac{2\sigma_{ij}(1-\sigma_{ij})}{\sqrt{d_C(i)}}h(\mathbf{x}_j)$$
$$=\frac{4\gamma}{\sqrt{d_C(i)}}\sum_{j=1}^{n}h(\mathbf{x}_j)\sigma_{ij}(1-\sigma_{ij})\left(\frac{2\sigma_{ij}-1}{\sqrt{d_C(i)}}\sum_{r\neq i}^{m}V_C(i,r)-\sum_{r\neq i}^{m}\frac{V_C(i,r)(2\sigma_{rj}-1)}{\sqrt{d_C(r)}}\right),$$

$$\frac{\partial C}{\partial\boldsymbol{\theta}_i}=-\sum_{j}^{n}[1-\sigma(Y_{ij}\boldsymbol{\theta}_i^Th(\mathbf{x}_j))]Y_{ij}h(\mathbf{x}_j),\quad \frac{\partial D}{\partial\boldsymbol{\theta}_i}=\eta\cdot\operatorname{diag}(\mathbf{u})^{-1}\boldsymbol{\theta}_i.$$

Combining the above four terms, i.e., $\nabla\boldsymbol{\theta}_i=\partial A/\partial\boldsymbol{\theta}_i+\partial B/\partial\boldsymbol{\theta}_i+\partial C/\partial\boldsymbol{\theta}_i+\partial D/\partial\boldsymbol{\theta}_i$, we obtain

$$\nabla\boldsymbol{\theta}_i=\eta\cdot\operatorname{diag}(\mathbf{u})^{-1}\boldsymbol{\theta}_i+2\sum_{j=1}^{n}h(\mathbf{x}_j)\cdot\sigma_{ij}(1-\sigma_{ij})\cdot\left(\frac{4\beta}{\sqrt{d_Y(j)}}\right.$$
$$\left.\cdot\left[\frac{2\sigma_{ij}-1}{\sqrt{d_Y(j)}}\sum_{r}^{n}V_Y(j,r)-\sum_{r}^{n}\frac{V_Y(j,r)(2\sigma_{ir}-1)}{\sqrt{d_Y(r)}}\right]+\frac{2\gamma}{\sqrt{d_C(i)}}\left[\frac{2\sigma_{ij}-1}{\sqrt{d_C(i)}}\right.\right.$$
$$\left.\left.\cdot\sum_{r\neq i}^{m}V_C(i,r)-\sum_{r\neq i}^{m}\frac{V_C(i,r)(2\sigma_{rj}-1)}{\sqrt{d_C(r)}}\right]\right)$$
$$-\sum_{j}^{n}[1-\sigma(Y_{ij}\boldsymbol{\theta}_i^Th(\mathbf{x}_j))]Y_{ij}h(\mathbf{x}_j). \tag{20}$$

Using $\nabla\boldsymbol{\theta}_i$, $\boldsymbol{\theta}_i$ can be updated by gradient descent with a line search based on Armijo rule. Then $\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_m$ are learned sequentially.

References

[1] Zhang Dengsheng, Md. Monirul Islam, Guojun Lu, A review on automatic image annotation techniques, Pattern Recognit. 45 (1) (2012) 346–362.
[2] Wallace V. Friesen Ekman, Paul, Joseph C. Hager, Facial Action Coding System (FACS): Manual, 1978.
[3] Grigorios Tsoumakas, Ioannis Katakis, Multi-label classification: an overview, Int. J. Data Warehous. Min. 3 (3) (2007) 1–13.
[4] Yu-Yin Sun, Yin Zhang, Zhi-Hua Zhou, Multi-label learning with weak label, in: AAAI, 2010.
[5] Serhat Selcuk Bucak, Rong Jin, Anil K. Jain, Multi-label learning with incomplete class assignments, in: CVPR, IEEE, Colorado, USA, 2011, pp. 2801–2808.
[6] Gang Chen, Yangqiu Song, Fei Wang, Changshui Zhang, Semi-supervised multi-label learning by solving a Sylvester equation, in: SIAM International Conference on Data Mining, 2008, pp. 410–419.
[7] Paul W. Holland, Roy E. Welsch, Robust regression using iteratively reweighted least-squares, Commun. Stat. Theory Methods 6 (9) (1977) 813–827.
[8] Luis Von Ahn, Laura Dabbish, Labeling images with a computer game, in: Proceedings of the SIGCHI Conference on Human factors in Computing Systems, ACM, Vienna, Australia, 2004, pp. 319–326.
[9] Patrick Lucey, Jeffrey F. Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, Iain Matthews, The extended Cohn-Kanade dataset (CK+): a complete dataset for action unit and emotion-specified expression, in: CVPR Workshops, IEEE, San Francisco, USA, 2010, pp. 94–101.
[10] Ashish Kapoor, Raajay Viswanathan, Prateek Jain, Multilabel classification using Bayesian compressed sensing, in: NIPS, 2012, pp. 2654–2662.
[11] Andrew B. Goldberg, Xiaojin Zhu, Ben Recht, Jun-Ming Xu, Robert D. Nowak, Transduction with matrix completion: three birds with one stone, in: NIPS, 2010, pp. 757–765.
[12] Ricardo Silveira Cabral, Fernando De la Torre, João Paulo Costeira, Alexandre Bernardino, Matrix completion for multi-label image classification, in: NIPS, 2011, pp. 190–198.
[13] Miao Xu, Rong Jin, Zhi-Hua Zhou, Speedup matrix completion with side information: application to multi-label learning, in: NIPS, 2013, pp. 2301–2309.
[14] Claudio Cusano, Gianluigi Ciocca, Raimondo Schettini, Image annotation using SVM, in: Electronic Imaging, International Society for Optics and Photonics, 2003, pp. 330–338.
[15] Qi Xiaojun, Yutao Han, Incorporating multiple SVMs for automatic image annotation, Pattern Recognit. 40 (2) (2007) 728–741.
[16] Olivier Chapelle, Patrick Haffner, Vladimir N. Vapnik, Support vector machines for histogram-based image classification, IEEE Trans. Neural Netw. 10 (5) (1999) 1055–1064.
[17] Raphael Maree, Pierre Geurts, Justus Piater, Louis Wehenkel, Random subwindows for robust image classification, in: CVPR, vol. 1, IEEE, San Diego, USA, 2005, pp. 34–40.
[18] Yasuhide Mori, Hironobu Takahashi, Ryuichi Oka, Image-to-word transformation based on dividing and vector quantizing images with words, in: The First International Workshop on Multimedia Intelligent Storage and Retrieval Management, Citeseer, 1999.
[19] Aditya Vailaya, Mário A.T. Figueiredo, Anil K. Jain, Hong-Jiang Zhang, Image classification for content-based indexing, IEEE Trans. Image Process. 10 (1) (2001) 117–130.
[20] Changbo Yang, Ming Dong, Farshad Fotouhi, Image content annotation using Bayesian framework and complement components analysis, in: ICIP, vol. 1, IEEE, Genoa, Italy, 2005, p. I-1193.
[21] Shi Rui, Wanjun Jin, Tat-Seng Chua, A novel approach to auto image annotation based on pairwise constrained clustering and semi-Naïve Bayesian model, in: MMM, IEEE, Melbourne, Australia, 2005, pp. 322–327.
[22] Jiwoon Jeon, Victor Lavrenko, Raghavan Manmatha, Automatic image annotation and retrieval using cross-media relevance models, in: SIGIR, ACM, Toronto, Canada, 2003, pp. 119–126.
[23] Changhu Wang, Shuicheng Yan, Lei Zhang, Hong-Jiang Zhang, Multi-label sparse coding for automatic image annotation, in: CVPR, IEEE, Miami, USA, 2009, pp. 1643–1650.
[24] Fei Wu, Yahong Han, Qi Tian, Yueting Zhuang, Multi-label boosting for image annotation by structural grouping sparsity, in: Proceedings of the International Conference on Multimedia, ACM, Firenze, Italy, 2010, pp. 15–24.
[25] Bihan Jiang, Michel François Valstar, Maja Pantic, Action unit detection using sparse appearance descriptors in space–time video volumes, in: FG Workshops, IEEE, Santa Barbara, California, USA, 2011, pp. 314–321.

Please cite this article as: B. Wu, et al., Multi-label learning with missing labels for image annotation and facial action unit recognition, Pattern Recognition (2015), http://dx.doi.org/10.1016/j.patcog.2015.01.022

[26] Michel François Valstar, Bihan Jiang, Marc Mehu, Maja Pantic, Klaus Scherer, The first facial expression recognition and analysis challenge, in: FG Workshops, IEEE, Santa Barbara, California, USA, 2011, pp. 921–926.
[27] Yunfeng Zhu, Fernando De la Torre, Jeffrey F. Cohn, Yu-Jin Zhang, Dynamic cascades with bidirectional bootstrapping for spontaneous facial action unit detection, in: The Third International Conference on Affective Computing and Intelligent Interaction and Workshops, IEEE, Amsterdam, The Netherlands, 2009, pp. 1–8.
[28] Michel F. Valstar, Maja Pantic, Combined support vector machines and hidden markov models for modeling facial action temporal dynamics, in: Human–Computer Interaction, Springer, Beijing, P.R. China, 2007, pp. 118–127.
[29] Tobias Gehrig, Hazim Kemal Ekenel, Facial action unit detection using kernel partial least squares, in: ICCV Workshops, IEEE, Barcelona, Spain, 2011, pp. 2092–2099.
[30] Yan Tong, Wenhui Liao, Qiang Ji, Facial action unit recognition by exploiting their dynamic and semantic relationships, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1683–1699.
[31] Yan Tong, Jixu Chen, Qiang Ji, A unified probabilistic framework for spontaneous facial action modeling and understanding, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2) (2010) 258–273.
[32] Arman Savran, BüLent Sankur, M. Taha Bilge, Comparative evaluation of 3D vs. 2D modality for automatic detection of facial action units, Pattern Recognit. 45 (2) (2012) 767–782.
[33] Yongqiang Li, Shangfei Wang, Yongping Zhao, Qiang Ji, Simultaneous facial feature tracking and facial expression recognition, IEEE Trans. Image Process. 22 (2013) 2559–2573.
[34] Zheng-Jun Zha, Xian-Sheng Hua, Tao Mei, Jingdong Wang, Guo-Jun Qi, Zengfu Wang, Joint multi-label multi-instance learning for image classification, in: CVPR, IEEE, Anchorage, AL, USA, 2008, pp. 1–8.
[35] Xiangyang Xue, Wei Zhang, Jie Zhang, Bin Wu, Jianping Fan, Yao Lu, Correlative multi-label multi-instance image annotation, in: ICCV, IEEE, Barcelona, Spain, 2011, pp. 651–658.
[36] Oksana Yakhnenko, Vasant Honavar, Multi-instance multi-label learning for image classification with large vocabularies, in: BMVC, 2011, pp. 1–12.
[37] Zhi-Hua Zhou, Min-Ling Zhang, Multi-instance multi-label learning with application to scene classification, in: NIPS, 2006, pp. 1609–1616.
[38] Min-Ling Zhang, Zhi-Hua Zhou, M3MIML: a maximum margin method for multi-instance multi-label learning, in: ICDM, IEEE, Pisa, Italy, 2008, pp. 688–697.
[39] Lihi Zelnik-Manor, Pietro Perona, Self-tuning spectral clustering, in: NIPS, 2004, pp. 1601–1608.
[40] Ulrike Von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.
[41] Yi Yang, Heng Tao Shen, Zhigang Ma, Zi Huang, Xiaofang Zhou, L2, 1-norm regularized discriminative feature selection for unsupervised learning, in: IJCAI, vol. 2, AAAI Press, Barcelona, Spain, 2011, pp. 1589–1594.
[42] Larry Armijo, Minimization of functions having Lipschitz continuous first partial derivatives, Pac. J. Math. 16 (1) (1966) 1–3.
[43] Mark J. Huiskes, Michael S. Lew, The MIR Flickr retrieval evaluation, in: Proceedings of the International Conference on Multimedia Information Retrieval, ACM, Vancouver, Canada, 2008, pp. 39–43.
[44] Xiangyu Chen, Yadong Mu, Shuicheng Yan, Tat-Seng Chua, Efficient large-scale image annotation by probabilistic collaborative multi-label propagation, in: Proceedings of the International Conference on Multimedia, ACM, Firenze, Italy, 2010, pp. 35–44.
[45] Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek, Cordelia Schmid, TagProp: discriminative metric learning in nearest neighbor models for image auto-annotation, in: ICCV, 2009, pp. 309–316.
[46] Yin Zhang, Zhi-Hua Zhou, Multilabel dimensionality reduction via dependence maximization, ACM Trans. Knowl. Discov. Data 4 (3) (2010) 14.
[47] Chih-Chung Chang, Chih-Jen Lin, Libsvm: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 27.
[48] Xiaodi Hou, Liqing Zhang, Saliency detection: a spectral residual approach, in: CVPR, IEEE, Minneapolis, MN, USA, 2007, pp. 1–8.
[49] Geoffrey E. Hinton, Simon Osindero, Yee-Whye Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[50] Min-Ling Zhang, Kun Zhang, Multi-label learning by exploiting label dependency, in: SIGKDD, ACM, Washington, DC, USA, 2010, pp. 999–1008.
[51] Xiangyu Chen, Xiao-Tong Yuan, Qiang Chen, Shuicheng Yan, Tat-Seng Chua, Multi-label visual classification with label exclusive context, in: ICCV, 2011, pp. 834–841.
[52] Arman Savran, Bulent Sankur, M. Taha Bilge, Regression-based intensity estimation of facial action units, Image Vis. Comput. 30 (10) (2012) 774–784.

Baoyuan Wu received his B.S. degree in Automation from University of Science and Technology Beijing, China, in 2009, and his Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2014. From September 2011 to September 2013, he was a Visiting Student in Rensselaer Polytechnic Institute, Troy, NY, USA. He is currently a Post-Doctoral Research Associate in King Abdullah University of Science and Technology. His main research interests include probabilistic graphical models and multi-label learning and clustering.

Siwei Lyu received his B.S. degree (Information Science), in 1997, and his M.S. degree (Computer Science), in 2000, both from Peking University, China. He received his Ph.D. degree in Computer Science from Dartmouth College, in 2005. From 2000 to 2001, he worked at Microsoft Research Asia (then Microsoft Research China) as an Assistant Researcher. From 2005 to 2008, he was a Post-Doctoral Research Associate at the Howard Hughes Medical Institute and the Center for Neural Science of New York University. Starting in 2008, he was an Assistant Professor at the Computer Science Department of University at Albany, State University of New York, and was promoted to Associate Professor in 2014. He is the recipient of the Alumni Thesis Award of Dartmouth College in 2005, IEEE Signal Processing Society Best Paper Award in 2010, and the NSF CAREER Award in 2010. He has authored one book, and held two U.S. and one E.U. patents. He has published more than 50 conference and journal papers in the research fields of natural image statistics, digital image forensics, machine learning and computer vision.

Bao-Gang Hu received his M.Sc. degree from the University of Science and Technology Beijing, China, in 1983, and his Ph.D. degree from McMaster University, Canada, in 1993, all in Mechanical Engineering. From 1994 to 1997, He was a Research Engineer and Senior Research Engineer at C-CORE, Memorial University of Newfoundland, Canada. Currently, he is a Professor with NLPR (National Laboratory of Pattern Recognition), Institute of Automation, Chinese Academy of Science, Beijing, China. From 2000 to 2005, he was the Chinese Director of LIAMA (the Chinese-French Joint Laboratory for Computer Science, Control and Applied Mathematics). His main research interests include intelligent systems, pattern recognition, and plant growth modeling. He is a Senior Member of IEEE.

Qiang Ji received his Ph.D. degree in Electrical Engineering from the University of Washington. He is currently a Professor with the Department of Electrical, Computer, and Systems Engineering at Rensselaer Polytechnic Institute (RPI). From 2009 to 2010, he served as a Program Director at the National Science Foundation (NSF), where he managed NSF's computer vision and machine learning programs. He also held teaching and research positions with the Beckman Institute at University of Illinois at Urbana-Champaign, the Robotics Institute at Carnegie Mellon University, the Department of Computer Science at University of Nevada at Reno, and the US Air Force Research Laboratory. He currently serves as the director of the Intelligent Systems Laboratory (ISL) at RPI. His research interests are in computer vision, probabilistic graphical models, pattern recognition, and their applications in various fields. He has published over 200 papers in peer-reviewed journals and conferences, and he has received multiple awards for his work. His research has been supported by major governmental agencies including NSF, NIH, DARPA, ONR, ARO, and AFOSR as well as by major companies. He is an Editor on several related IEEE and international journals and he has served as a general chair, program chair, technical area chair, and program committee member in numerous international conferences/workshops. He is a Fellow of the IEEE and the IAPR.

Please cite this article as: B. Wu, et al., Multi-label learning with missing labels for image annotation and facial action unit recognition, Pattern Recognition (2015), http://dx.doi.org/10.1016/j.patcog.2015.01.022