Pattern Recognition Letters 34 (2013) 1964–1970

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

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

Learning person-specific models for facial expression and action unit recognition

Jixu Chen [a,*], Xiaoming Liu [b], Peter Tu [a], Amy Aragones [a]

[a] GE Global Research, Niskayuna, NY 12309, United States
[b] Michigan State University, East Lansing, MI, 48824, United States

ARTICLE INFO

Article history:
Available online 19 February 2013

Keywords:
Transfer learning
Expression recognition
Action unit recognition

ABSTRACT

A key assumption of traditional machine learning approach is that the test data are draw from the same distribution as the training data. However, this assumption does not hold in many real-world scenarios. For example, in facial expression recognition, the appearance of an expression may vary significantly for different people. As a result, previous work has shown that learning from adequate person-specific data can improve the expression recognition performance over the one from generic data. However, person-specific data is typically very sparse in real-world applications due to the difficulties of data collection and labeling, and learning from sparse data may suffer from serious over-fitting. In this paper, we propose to learn a person-specific model through transfer learning. By transferring the informative knowledge from other people, it allows us to learn an accurate model for a new subject with only a small amount of person-specific data. We conduct extensive experiments to compare different person-specific models for facial expression and action unit (AU) recognition, and show that transfer learning significantly improves the recognition performance with a small amount of training data.

1. Introduction

In recent years, machine learning approaches have been successfully applied to the field of human action recognition, including automatic facial expression recognition. Traditionally, many machine learning algorithms work well only under the assumption that the training and test data are drawn from the same distribution. In facial expression recognition, this assumption holds for some prototypical and posed expressions, such as the "smiling" faces from the Cohn–Kanade DFAT database (Kanade et al., 2000) (Fig. 1(a)). Because smile is quite consistent across subjects, the state-of-the-art smile detection system can easily achieve an accuracy of 97% (Whitehill et al., 2009) on the DFAT database via leave-one-subject-out cross validation. However, the identical-distribution assumption does not hold for complex and spontaneous expressions. For example, the PAINFUL database (Lucey et al., 2011a) contains the spontaneous pain expressions of patients with shoulder injury when they move their shoulders, as shown in Fig. 1(b). We can observe large variation of the pain expression across different subjects, such as eyes open or closed, mouth open or closed, etc. Because the training and test data may not share the same distribution, the performance of the pain detection is much worse than that of the smile detection.

When the appearance of the facial expression changes across the subjects, learning a person-specific model is likely to achieve better performance than a generic model. However, in many real-world applications, it is not only expensive to collect and label a large amount of data for a specific person, but also impractical in some scenarios. For example, in pain expression recognition, a new subject has to enact the pain expression specifically for the data collection. This process is unnatural and cumbersome for the subject, and this posed expression may be different from the spontaneous expression in the actual testing scenario. Thus, how to learn a person-specific model with limited person-specific data becomes a critical research problem.

In this paper, we exploit a new promising way to learn a person-specific model via transfer learning. Transfer learning represents a family of algorithms that transfer the informative knowledge from the source domain to a new target domain. In our applications, we view the data of the subject of interest as the target domain, and the training data of other subjects as the source domain. We consider two transfer learning scenarios: inductive transfer learning (Section 3.1) and transductive transfer learning (Section 3.2). For the former, only a small amount of labeled data from the target domain are required to learn the robust target model without overfitting. For the latter, the target data does not need to be labeled hence the burden of data labeling is entirely avoided.

We apply our algorithm to two recognition tasks: the aforementioned pain expression recognition and facial action unit

* Corresponding author. Tel.: +1 518 387 5567; fax: +1 518 387 4136.
E-mail address: chenji@ge.com (J. Chen).

0167-8655/$ - see front matter © 2013 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.patrec.2013.02.002

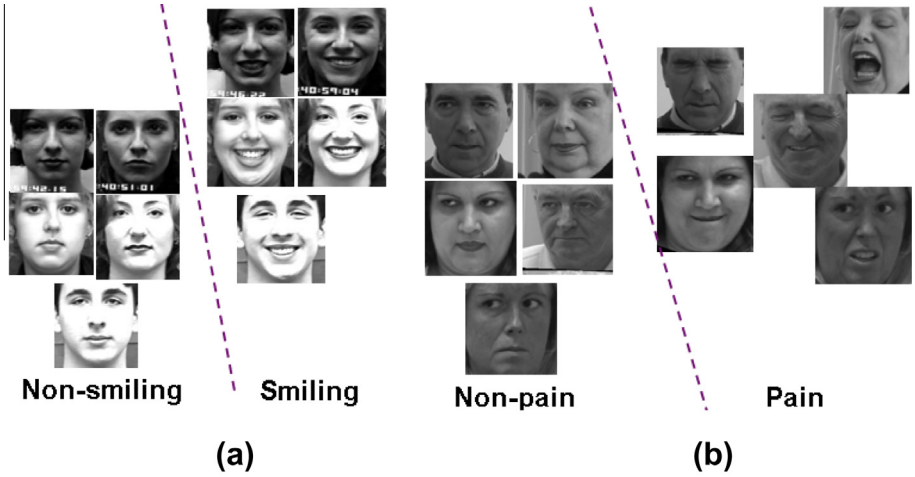

Fig. 1. (a) The smile expressions from the DFAT database (Kanade et al., 2000). (b) The spontaneous pain expressions from the PAINFUL database (Lucey et al., 2011a). Pain expression has large variation across subjects.

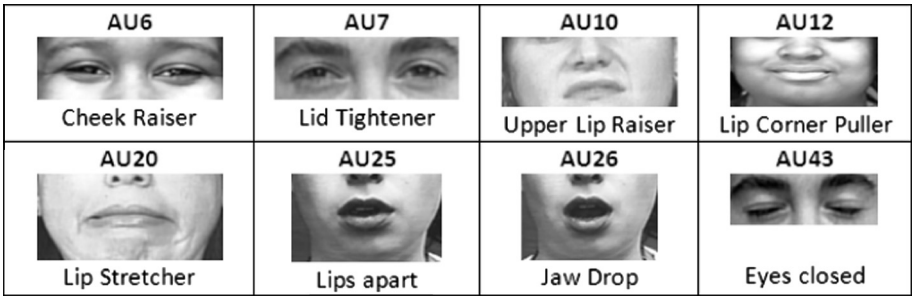

Fig. 2. Examples of facial action units.

recognition. Action units (AUs) are a set of local facial behavior descriptors defined in the widely used Facial Action Coding System (FACS) (Ekman and Friesen, 1978). Based on FACS, the facial behavior is decomposed into 46 AUs, where some frequent action units are shown in Fig. 2. Although only a small number of AUs are defined, over 7000 different AU combinations have been observed so far (Ekman and Friesen, 1978). Thus, AU recognition result can be treated as the recognition of more generalized expressions which are basically viewed as certain combinations of AUs.

We compare various transfer learning algorithms and traditional learning algorithms in our experiment and show significant improvement of the inductive transfer learning in both the expression recognition and the AU recognition (Section 4).

2. Related work

In recent years, facial expression recognition (e.g. happy, anger, disgust, fear, sadness, surprise) (Cohen et al., 2003) and FACS facial action units (AU) recognition (Ekman et al., 2005) have made considerable progress. A comprehensive review can be found in (Zeng et al., 2009). However, most of the current expression and AU recognition research has focused on the posed expression under tightly controlled laboratory conditions, e.g. Cohn–Kanade DFAT (Kanade et al., 2000), CMU-PIE (Sim et al., 2003) and MMI (Pantic et al., 2005) expression database. There have been very little work on detecting natural spontaneous facial expression (Tong et al., 2010; Bartlett et al., 2006) which varies significantly across subjects. The first attempt to the spontaneous expression recognition is on the RUFACS database (Bartlett et al., 2006), which consists of 34 subjects. They are asked to express an opinion on a social or political issue and convince an interviewer that they are telling the truth. This dataset contains many subtle facial expressions indicative of the natural human behavior, and these subtle expressions are different across subjects. Bartlett et al. (2006) found that the performance of spontaneous expression recognition diminished greatly when compared to the scenario of posed expression.

An application of spontaneous facial expression recognition that would be of great benefit is pain and no-pain classification (Lucey et al., 2011b). For instance, in intensive care units (ICU) (Gawande, 2009), the improvement in patient outcomes has been achieved by pain monitoring. Lucey et al. (2011b) collect the spontaneous pain database from patients with shoulder injuries. Their pain detection system achieved 0.751 area under the ROC curve (AUC) using only appearance features, and achieved the best performance of 0.839 AUC by combining the shape and appearance features. In this paper, we propose to further improve this state-of-the-art performance through a person-specific expression recognizer.

Previous work (Cohen et al., 2003; Valstar et al., 2011) has shown that a person-specific model out-performs a person-independent model in expression recognition when adequate person-specific data is available. Hence, in order to learn a person-specific pain detector deployed in a healthcare application, the doctor or nurse has to enroll and label the pain expressions for every to-be-detected patient. It would greatly reduce the data collection burden if only a small number of training images for each patient are required. For traditional machine learning algorithms, learning from a small amount of data can be exposed to the risk of overfitting. In this paper, we propose to learn the person-specific model through transfer learning.

Transfer learning aims to extract knowledge from one or more source domains and improve the learning in the target domain. It

has been applied to a wide variety of applications, such as object recognition (Yao and Doretto, 2010), sign language recognition (Farhadi et al., 2007) and text classification (Wang et al., 2008). For more details we refer the reader to the survey paper (Pan and Yang, 2010). In (Pan and Yang, 2010), the transfer learning algorithms are classified into three categories, namely *inductive transfer learning*, *transductive transfer learning* and *unsupervised transfer learning*.

In *inductive transfer learning* (Dai et al., 2007; Yang et al., 2007), only a small amount of labeled data in the target domain is available. Learning a classifier solely from the labeled target data may suffer serious overfitting. Transfer learning remedies this problem by using the knowledge from the data in the source domain. *TrAdaBoost* (Dai et al., 2007) attempts to utilize the "good" data in the source domain, which are similar to the target data, to improve the target Adaboost classifier. Kulis et al. (2011) propose a domain adaption approach for object recognition. From the labeled object categories, they learn a non-linear transformation to transfer the data points in the source domain to the target domain.

In *transductive transfer learning* (Zadrozny, 2004; Huang et al., 2006; Sugiyama et al., 2007; Si et al., 2010), the target data is available but not labeled. Only the source data has labels. Thus, we cannot learn a classifier directly from the unlabeled target data. A common approach is to shift or re-weight the labeled source data, from which a target classifier can be learned. In (Zadrozny, 2004; Huang et al., 2006), the source training data is re-weighted to approximate the distribution in the target domain. Gopalan et al. (2011) propose to learn a domain shift from the source subspace to the target subspace in Grassmann manifold, and project the labeled source data to a subspace close to the target domain. Another approach for transductive transfer learning is to incorporate the unlabeled target data in the training of the source domain. Si et al. (2010) propose to use the unlabeled target data as a regularization term in the discriminative subspace learning in the source domain, so that the learned subspace can generalize well to the target domain. Please notice that the term "transductive transfer learning" was first proposed by Arnold et al. (2007) to distinguish it from "transductive learning" (Vapnik et al., 1995) in the traditional machine learning setting. In transductive learning (Joachims, 1999; Li and Wechsler, 2005), the unlabeled testing data is known at the training stage, which allows the learner to shape its decision function to match the properties of testing data. However, in transductive learning, the training and testing data are assumed to be drawn from the same distribution, while in transductive transfer learning, the source and target data are drawn from different distributions.

Finally, the *unsupervised transfer learning* (Dai et al., 2008) is applied to unsupervised learning tasks, such as clustering and dimensionality reduction, when both the target labels and the source labels are not available. In this paper, we apply both the inductive and transductive transfer learning to the task of person-specific facial expression and AU recognition.

3. Learning a person-specific model

We first introduce the notation used in our transfer learning problem. Let's denote the training data of a new subject as the target data $\mathbf{D}_T = \{(\mathbf{x}_{T,i}, y_{T,i})\}_{i=1\cdots N_T}$ and the training data of other M subjects as the source data $\mathbf{D}_S = \{\mathbf{D}_1, \ldots, \mathbf{D}_M\}$, where $\mathbf{D}_m = \{(\mathbf{x}_{m,1}, y_{m,1}), \ldots, (\mathbf{x}_{m,N_m}, y_{m,N_m})\}$, $\mathbf{x} \in \mathcal{X}$ is in the feature space and $y \in \{1, +1\}$ is the binary class label. For example, in expression recognition, y represents the presence or absence of certain facial expression. A person-specific model is a classifier $f_T : \mathbf{x}_T \to y_T$ learned from the target data $\mathbf{D}_T$. However, since the size of target data (N_T) is very small, learning from $\mathbf{D}_T$ alone may suffer serious overfitting problems.

The most straightforward approach to address this problem is to combine the source data with the person-specific target data, and learn the classifier f_T from $\{\mathbf{D}_T, \mathbf{D}_S\}$. In the first facial expression recognition and analysis (FERA) challenge (Valstar et al., 2011), this method works well for the expression recognition task. For example, the F-score is 0.44 for the person-independent test, and it is improved to 0.73 for the person-specific test. However, this method may have problems when the size of the target data is much smaller than that of the source data, i.e., $N_T \ll N_S$. In this case, the target data is likely to be ignored in the combined training data set, because data samples from $\mathbf{D}_T$ and $\mathbf{D}_S$ contribute equally to the learning.

In comparison, transfer learning focuses on the performance on the target data. It can improve the learning of f_T by transferring informative knowledge from the abundant source data $\mathbf{D}_S$. According to Pan and Yang (2010), there are two types of transfer learning algorithms that are suitable for learning f_T.

3.1. Inductive transfer learning algorithm

In this section, we use the boosting-based inductive transfer learning in (Yao and Doretto, 2010) to learn a person-specific model. This framework consists of two phases. In the first phase, the knowledge of the source data is represented by a large collection of weak classifiers. In the second phase, some of the weak classifiers are selected to boost the classification performance on the target data.

Algorithm 1. Inductive transfer learning for a person-specific model

input: Source data of M subjects $\mathbf{D}_1, \ldots, \mathbf{D}_M$ and target data of a subject $\mathbf{D}_T$.
output: A person-specific classifier for the target subject $y = f_T(\mathbf{x})$.
Phase-I Learning a weak classifier set $\mathcal{H} = \{h_m^{(k)}\}$ from source data $\mathbf{D}_1, \ldots, \mathbf{D}_M$.
for $m = 1$ to M **do**
  Initialize the weight vector $\mathbf{w}_m^{(1)} = (w_{m,1}^{(1)}, \ldots, w_{m,N_m}^{(1)})$,
  for $k = 1$ to K **do**
    Normalize the weight vector $\mathbf{w}_m$ to 1,
    Find $h_m^{(k)}$ that minimizes the weighted classification error ε of $\mathbf{D}_m$,
    Compute the weighted error $\varepsilon = \sum_{i=1}^{N_m} w_{m,i}^{(k)}[y_{m,i} \neq h_m^{(k)}(\mathbf{x}_{m,i})]$,
    $\alpha = \frac{1}{2}\ln\frac{1-\varepsilon}{\varepsilon}$,
    Update the weights $w_{m,i}^{(k+1)} = w_{m,i}^{(k)} \exp\{-\alpha y_{m,i} h_m^{(k)}(\mathbf{x}_{m,i})\}$,
    $\mathcal{H} \leftarrow \mathcal{H} \bigcup h_m^{(k)}$.
  end for
end for
Phase-II Learning a target classifier on target data $\mathbf{D}_T$.
Initialize the weights $\mathbf{w}_T^{(1)} = (w_{T,1}^{(1)}, \ldots, w_{T,N_T}^{(1)})$,
for $k = 1$ to K **do**
    Normalize the weight vector $\mathbf{w}_T$ to 1,
    Select $h_T^{(k)}$ from $\mathcal{H}$ that minimizes the weighted classification error ε of $\mathbf{D}_T$,
    Compute the weighted error $\varepsilon = \sum_{i=1}^{N_T} w_{T,i}^{(k)}[y_{T,i} \neq h_T^{(k)}(\mathbf{x}_{T,i})]$,
    $\alpha_T^{(k)} = \frac{1}{2}\ln\frac{1-\varepsilon}{\varepsilon}$,
    Update the weights $w_{T,i}^{(k+1)} = w_{T,i}^{(k)} \exp\{-\alpha_T^{(k)} y_{T,i} h_T^{(k)}(\mathbf{x}_{T,i})\}$,
    $\mathcal{H} \leftarrow \mathcal{H} \setminus h_T^{(k)}$.
end for
return $f_T(\mathbf{x}) = sign\left(\sum_k \alpha_T^{(k)} h_T^{(k)}(\mathbf{x})\right)$.

The transfer learning algorithm is summarized in Algorithm 1. Notice that it transfers the knowledge from multiple sources, each is the training data of one subject. The total number of the source data samples is $N_S = \sum_{m=1}^{M} N_m$. Compared to the transfer learning from a single source (Dai et al., 2007), this multi-source transfer learning is able to identify and take advantage of the sources that are closely related to the target, making it less vulnerable to *negative transfer* from the unrelated sources.

Phase-I is the standard Adaboost algorithm conducted for each subject of the source data. The Adaboost classifier includes the weak classifiers that best discriminate the positive and negative data for a particular source subject. All the weak classifiers learned from source data constitute a large set of classifiers $\mathcal{H}$. Phase-II is a variation of Adaboost on the target data $\mathbf{D}_T$. In contrast to the traditional Adaboost which learns weak classifiers from the target data, we select the weak classifiers from the source classifier set $\mathcal{H}$, which basically stores all weak classifiers work well for the source data. Since only the classifiers with the lowest classification rate on $\mathbf{D}_T$ are finally selected, it ensures the *positive transfer* of the knowledge from the source domain to the target domain.

3.2. Transductive transfer learning algorithm

In this section, we use the transductive transfer learning algorithm in (Sugiyama et al., 2007) to learn a person-specific model. This approach is attractive because it can learn the target classifier without knowing the target labels $\{y_{T,1}, \ldots, y_{T,N_T}\}$, so that the burden of manual labeling for a new subject can be entirely eliminated.

The basic idea of transfer learning is to re-use the source data that is close to the target. Given the labeled source data $\mathbf{D}_S = \{(\mathbf{x}_{S,i}, y_{S,i})\}_{i=1\ldots N_S}$ and the unlabeled target data $\mathbf{D}_T = \{\mathbf{x}_{T,j}\}_{j=1\ldots N_T}$, transductive transfer learning reweights every sample $(\mathbf{x}_{S,i}, y_{S,i})$ in the source data using the probability ratio $w(\mathbf{x}_{S,i}) = \frac{p_S(\mathbf{x}_{S,i})}{p_T(\mathbf{x}_{S,i})}$, where $p_S(\mathbf{x})$ and $p_T(\mathbf{x})$ are the marginal distributions of the source and the target, and then the reweighted source data are used to train the target model.

Here, the sample weight $w(\mathbf{x})$ is approximated by a linear model,

$$\hat{w}(\mathbf{x}) = \sum_{l=1}^{b} \alpha_l \phi_l(\mathbf{x}), \tag{1}$$

where $\phi_l(\mathbf{x})$ is a basis function such that $\phi_l(\mathbf{x}) \geqslant 0$ for all $\mathbf{x}$, and α_l is the parameter to be learned. In our experiment, we use the kernel function as the basis function: $\phi_l(\mathbf{x}) = \mathbf{K}(\mathbf{x}, \mathbf{x}_l) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_l\|}{2\sigma^2}\right)$, where σ is the kernel width; $\mathbf{x}_l$ is a data sample randomly selected from the target data. We randomly select half of the target data set to estimate these basis functions.

Based on Eq. (1), the target distribution can be approximated by the weighted source distribution,

$$\hat{p_T}(\mathbf{x}) = \hat{w}(\mathbf{x}) p_S(\mathbf{x}). \tag{2}$$

Transductive transfer learning minimizes the KullbackLeibler (KL) divergence between $\hat{p_T}(\mathbf{x})$ and $p_T(\mathbf{x})$, with respect to $\{\alpha_l\}_{l=1}^{b}$,

$$KL[p_T(\mathbf{x})||\hat{p_T}(\mathbf{x})] = \int p_T(\mathbf{x}) \log \frac{p_T(\mathbf{x})}{\hat{w}(\mathbf{x}) p_S(\mathbf{x})} d\mathbf{x} = \int p_T(\mathbf{x}) \log \frac{p_T(\mathbf{x})}{p_S(\mathbf{x})} d\mathbf{x} - \int p_T(\mathbf{x}) \log \hat{w}(\mathbf{x}) d\mathbf{x}. \tag{3}$$

Given the training data, the first term is independent of $\{\alpha_l\}_{l=1}^{b}$. Thus, we maximize the second term,

$$J := \int p_T(\mathbf{x}) \log \hat{w}(\mathbf{x}) d\mathbf{x} \approx \frac{1}{n_T} \sum_j \log \hat{w}(\mathbf{x}_{T,j}) = \frac{1}{n_T} \sum_j \log \left(\sum_{l=1}^{b} \alpha_l \phi_l(\mathbf{x}_{T,j}) \right), \tag{4}$$

subject to the constraint,

$$1 = \int \hat{w}(\mathbf{x}) p_S(\mathbf{x}) d\mathbf{x} = \frac{1}{n_S} \sum_i \hat{w}(\mathbf{x}_{S,i}) = \frac{1}{n_S} \sum_i \sum_{l=1}^{b} \alpha_l \phi_l(\mathbf{x}_{S,i}). \tag{5}$$

For the details of this optimization procedure, please refer to Sugiyama et al. (2007). After optimization, we can used the estimated $\{\alpha_l\}_{l=1}^{b}$ to compute the weight of each source data sample using Eq. (1).

Finally, we use the weighted source data to train an Adaboost classifier for the target subject, i.e., the sample weights of the source data are initialized as $\{\hat{w}(\mathbf{x}_{S,i})_{i=1\ldots N_S}\}$ in the AdaBoost learning algorithm.

4. Experimental results

We apply the transfer learning algorithms in two experiments, namely pain expression recognition and facial action unit (AU) recognition. Pain is an holistic expression as shown in Fig. 1(b), which is of specific interest in health care applications (Gawande, 2009). Facial action units are a set of local facial muscular movements defined by psychologists. Over 7000 different facial expressions can be represented by the combinations of 46 AUs (Ekman and Friesen, 1978). By applying transfer learning to AU recognition, we expect to generalize our algorithm to the recognition of more facial expressions.

We test the transfer learning algorithms on the PAINFUL database (Lucey et al., 2011a), which contains video sequences (totally 48,398 frames) of 25 patients with shoulder injuries. Spontaneous pain facial expressions are captured when the patients are rotating their arms.

4.1. Feature extraction

Local Binary Pattern (LBP) (Ahonen et al., 2006) is used as the facial image feature in our experiments. We first use the eye locations provided in the PAINFUL database to crop and warp the face region to a 128×128 image. Following the similar method in (Ahonen et al., 2006), the face image is divided into 8×8 small regions to extract the $\text{LBP}^{u2}_{8,1}$ feature (59 dimensions). Here the superscript $u2$ reflects the use of uniform patterns, and (8,1) represents 8 sampling point on a circle of radius of 1. These LBP features are concatenated into a single, spatially enhanced feature which therefore has $59 \times 8 \times 8 = 3776$ dimensions, as shown in Fig. 3.

4.2. Pain expression recognition

Parkachin and Solomon pain intensity (PSPI) (Prkachin and Solomon, 2008) is defined for each frame in the PAINFUL database, with PSPI ranges from 0 to 16. Here we label the frames with PSPI $= 0$ as the negative samples and the frames with PSPI > 0 as the positive samples.

Similar to Lucey et al. (2011a), we perform a leave-one-subject-out evaluation on 25 subjects. The number of frames varies from 518 to 3360 for different subjects. For the target subject, we concatenate his positive frames into a sequence of length N_P and select the first N'_P frames as the target data to train a person-specific model. The second half of this sequence is retained for testing. Similarly, in N_N negative frames, first N'_N frames are selected as the target data and the last $\lfloor N_N/2 \rfloor$ frames are used for testing. Totally

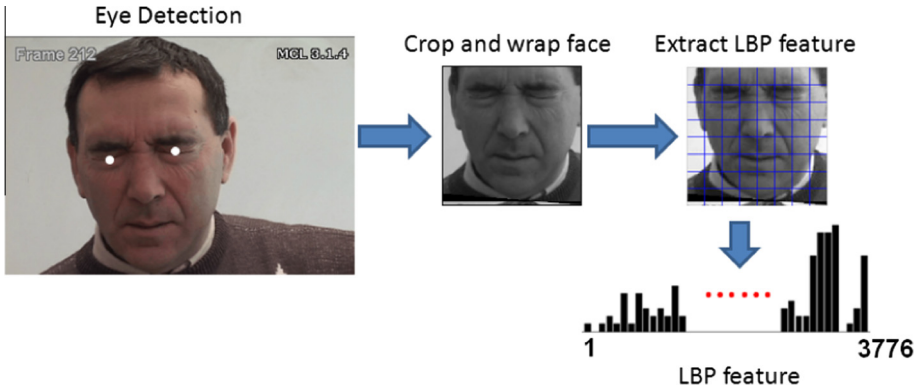

Fig. 3. LBP is extracted as the facial image feature.

$N'_T = N'_P + N'_N$ frames are used as the target data for transfer learning. We keep the positive/negative ratios to be the same $\frac{N_P}{N_N} = \frac{N'_P}{N'_N}$.

We compare the generic model and four different person-specific models as follows.

- Generic model is an Adaboost classifier learned from the source data of 24 subjects. This is our baseline algorithm;
- Traditional person-specific Model-A is an Adaboost classifier learned only from the target data without transfer learning;
- Traditional person-specific Model-B is an Adaboost classifier learned from a combined dataset of both the source data and the target data;
- Inductive transfer model is learned using Algorithm 1. The target data and their labels are used to select weak classifiers from the source classifier set;
- Transductive transfer model is learned using the algorithm in Section 3.2. The target data are used to reweight the source data. The labels of the target data are not used.

Each of the above models consists of 50 weak classifiers. When the number of training samples is 50, i.e., $N'_T = 50$, the ROCs of these models are shown in Fig. 4. Table 1 shows the area under ROC (AUC) of these models with different number of samples in the target data.

For the traditional person-specific Model-A, we learn the Adaboost classifier only from the person-specific target data. This model suffers serious overfitting problems when the target data is limited (AUC is 0.557 when the number of target data is 10). Its performance can be improved by adding more training data, but it is always worse than the inductive transfer learning. When using adequate training data (i.e., half of the sequence), its performance is close to the inductive transfer learning.

Table 1
Performance comparison: AUC for different sizes of target data (N'_T). 'Half' refers to using the first half of the sequence for transfer learning.

N'_T	10	25	50	100	Half
Traditional Model-A	0.557	0.684	0.786	0.862	0.893
Traditional Model-B	0.786	0.816	0.819	0.835	0.878
Inductive transfer	0.782	0.821	0.880	0.891	0.895
Transductive transfer	0.756	0.755	0.765	0.756	0.760
Generic model (baseline)	0.769				

Table 2
Average time for training a person-specific model.

Traditional Model-A	2.6 min
Traditional Model-B	14.3 min
Inductive transfer model	0.16 min
Transductive transfer model	17.6 min

For the traditional person-specific Model-B, the classifier is learned from a combined dataset that consists of both the source and the target data. Because we have a large amount of training data, the overfitting problem can be remedied. However, since this classifier focuses on the combined dataset, its performance on the target data is not as good as Model-A when the target data is sufficient. Furthermore, since the training data size is very large, the learning process is very time consuming. We list the average training time for various person-specific models in Table 2.

The inductive transfer learning achieves the best performance among person-specific models. It outperforms the baseline with a

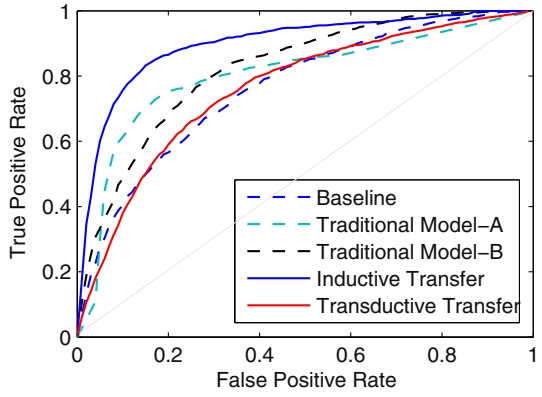

Fig. 4. ROCs when $N'_T = 50$.

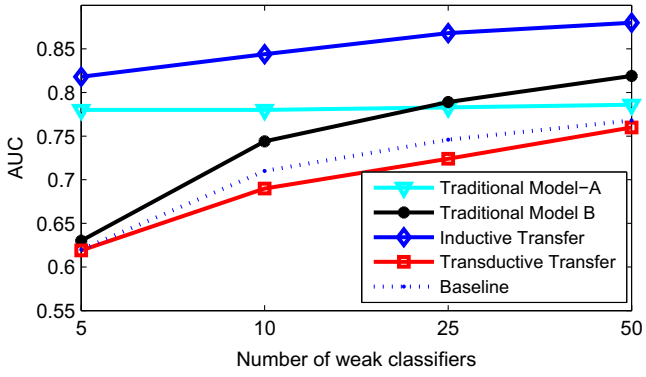

Fig. 5. AUC of different models with different numbers of weak classifiers when $N'_T = 50$.

Table 3
Comparison of AUCs of AU recognition. $N'_T = 50$ is used in inductive transfer learning.

AU	6	7	10	12	20	25	26	43	AVG
Generic model	0.792	0.634	0.758	0.772	0.820	0.655	0.596	0.875	0.738
Inductive transfer	0.907	0.921	0.930	0.901	0.888	0.855	0.864	0.922	0.8985
Lucey et al. (2011a)	0.854	0.804	0.892	0.857	0.779	0.780	0.710	0.875	0.8189

small number of target data samples (AUC is improved from 0.769 to 0.782 with only 10 samples) and its performance increases significantly when adding more training samples (AUC = 0.891 with 100 target samples). Furthermore, because inductive transfer learning does not need to train new weak classifiers, it is the fastest algorithm as shown in Table 2, which makes it suitable for rapid retraining of a new target subject.

For the transductive transfer learning, we did not observe any improvement even with adequate training data. A possible reason is that the boosting classifier is not sensitive to the marginal distribution change. In (Zadrozny, 2004), the classifiers are grouped into two categories: *local classifiers*, which depend only on $P(y|\mathbf{x})$, and *global classifiers*, which depend on both $P(y|\mathbf{x})$ and $P(\mathbf{x})$. In our transductive transfer learning, we only reweight the source data to approximate the target marginal distribution $P_T(\mathbf{x})$. Since the AdaBoost classifier tends to be a local learner, this transductive transfer scheme may not be suitable.

The training and testing time of an Adaboost classifier is proportional to its number of weak classifiers. An efficient algorithm can learn a good AdaBoost classifier with fewer weak classifiers. Fig. 5 depicts the performance of different algorithms with different numbers of weak classifiers. It shows that inductive transfer learning can achieve good performance (AUC = 0.818) with merely five weak classifiers, which further demonstrates its efficacy.

4.3. Action unit recognition

In this section we focus on AU recognition because we expect to generalize our transfer-learning framework to more expressions, which basically can be viewed as certain combinations of AUs. In the PAINFUL database, we focus on eight frequent AUs which are labeled frame by frame. The descriptions of these AUs are showed in Fig. 2. For each AU, we label the frames with AU presence as positive samples, and the other frames as negative samples. We use the same method in Section 4.1 to extract the 3776 dimensional LBP feature from each frame.

A leave-one-subject-out evaluation on 25 subjects is performed, using the same method in Section 4.2 to select target data and testing data for each subject. We compare the generic model, the inductive transfer model ($N'_T = 50$), and the state-of-the-art method (Lucey et al., 2011a) in Table 3. In (Lucey et al., 2011a), both face shape and appearance from Active Appearance Model (AAM) tracking are used as the image feature and a generic SVM classifier is trained for each AU. In our generic model and inductive transfer model, we only use the face appearance feature, i.e., the LBP feature. Each model is composed of 50 weak classifiers.

We observe significant performance improvement of inductive transfer learning given only a small number of target data ($N'_T = 50$). It outperforms our generic model and also the start-of-the-art model in (Lucey et al., 2011a). In terms of individual AU detection, transfer learning improves the performance of every AU, especially the AUs with small muscular movement, which are typically very difficult to be detected, such as eyelid tightening (AU7) and lip stretcher (AU20).

5. Conclusion

This paper exploits the idea of learning a person-specific model to improve facial expression recognition. In order to learn a robust person-specific model with minimal demand on new target data, we propose to use the transfer learning, which can mitigate the overfitting in the target domain by transferring the informative knowledge from similar source domains. We deploy and evaluate different transfer learning algorithms within the context of pain expression recognition and face AU recognition. Compared to the traditional methods, the experiment shows that *inductive transfer learning* can significantly improve the recognition performance with a limited number of target samples, through a very efficient learning procedure. In order to further reduce the dependence on data collection and labeling, our future work includes extending the algorithm to make use of the unlabeled data samples or partially labeled data, e.g., only the negative data is available and labeled.

References

Kanade, T., Cohn, J., Tian, Y.L., 2000. Comprehensive database for facial expression analysis. In: Proc. Internat. Conf. on Automatic Face and Gesture Recognition (FG), pp. 46–53.

Whitehill, J., Littlewort, G., Fasel, I., Bartlett, M., Moellan, J., 2009. Toward practical smile detection. IEEE Trans. Pattern Anal. Machine Intell. 31 (11), 2106–2111.

Lucey, P., Cohn, J.F., Prkachin, K.M., Solomon, P.E., Matthews, I., 2011. Painful data: The UNBC-McMaster shoulder pain expression archive database. In: Proc. Internat. Conf. on Automatic Face and Gesture Recognition (FG), pp. 57–64.

Ekman, P., Friesen, W.V., 1978. Facial Action Coding System: A Technique for the Measurement of Facial Moment. Consulting Psychologists Press.

Cohen, I., Sebe, N., Garg, A., Chen, L.S., Huang, T.S., 2003. Facial expression recognition from ideo sequences: Temporal and static modeling. Comput. Vision Image Understan. 91 (1–2), 160–187.

Ekman, P., Rosenberg, E., 2005. What the Face Reals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS). Series in Affective Science. Oxford University Press.

Zeng, Z., Pantic, M., Roisman, G.I., Huang, T.S., 2009. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. IEEE Trans. Pattern Anal. Machine Intell. 31 (1), 39–58.

Sim, T., Baker, S., Bsat, M., 2003. The CMU pose, illumination, and expression database. IEEE Trans. Pattern Anal. Machine Intell. 25 (1), 1615–1618.

Pantic, M., Valstar, M., Rademaker, R., Maat, L., 2005. Web-based database for facial expression analysis. In: Proc. Internat. Conf. on Multimedia and Expo (ICME).

Tong, Y., Chen, J., Ji, Q., 2010. A unified probabilistic framework for spontaneous facial action modeling and understanding. IEEE Trans. Pattern Anal. Machine Intell. 32 (2), 258–273.

Bartlett, M., Littlewort, G., Frank, M., Lainscsek, C., Fasel, I., Moellan, J., 2006. Automatic recognition of facial actions in spontaneous expressions. J. Multimedia 1 (6), 22–35.

Lucey, P., Cohn, J.F., Matthews, I., Lucey, S., Sridharan, S., Howlett, J., et al., 2011b. Automatically detecting pain in ideo through facial action units. IEEE Trans. Systems Man Cybern. Part B Cybern. 41 (3), 664–674.

Gawande, A., 2009. The Checklist Manifesto: How to Get Things Right. Metropolitan Books.

Valstar, M., Jiang, B., Mehu, M., Pantic, M., Scherer, K., 2011. The first facial expression recognition and analysis challenge. In: Proc Internat. Conf. on Automatic Face and Gesture Recognition (FG), pp. 921–926.

Yao, Y., Doretto, G., 2010. Boosting for transfer learning with multiple sources. In: Proc. IEEE Conf. on Computer Vision and, Pattern Recognition (CVPR), pp. 1855–1862.

Farhadi, A., Forsyth, D., White, R., 2007. Transfer learning in sign language. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 1–8.

Wang, P., Domeniconi, C., Hu, J., 2008. Using wikipedia for co-clustering based cross-domain text classification. In: Proc. Internat. Conf. on Data Mining (ICDM), pp. 1085–1090.

Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22 (10), 1345–1359.

Dai, W., Yang, Q., Xue, G.R., Yu, Y., 2007. Boosting for transfer learning. In: Proc. Internat. Conf. on, Machine Learning (ICML), pp. 193–200.
Yang, J., Yan, R., Hauptmann, A.G., 2007. Cross-domain ideo concept detection using adaptive sms. In: MULTIMEDIA, pp. 188–197.
Kulis, B., Saenko, K., Darrell, T., 2011. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: Proc. IEEE Conf. on Computer Vision and, Pattern Recognition (CVPR), pp. 1785–1792.
Zadrozny, B., 2004. Learning and evaluating classifiers under sample selection bias. In: Proc. Internat. Conf. on, Machine Learning (ICML), pp. 903–910.
Huang, J., Smola, A.J., Gretton, A., Borgwardt, K.M., Schlkopf, B., 2006. Correcting sample selection bias by unlabeled data. Adv. Neural Inf. Process. Systems (NIPS), 601–608.
Sugiyama, M., Nakajima, S., Kashima, H., von Bünau, P., Kawanabe, M., 2007. Direct importance estimation with model selection and its application to covariate shift adaptation. Adv. Neural Inf. Process. Systems (NIPS) 20, 1433–1440.
Si, S., Tao, D., Geng, B., 2010. Bregman divergence-based regularization for transfer subspace learning. IEEE Trans. Knowl. Data Eng. 22, 929–942.
Gopalan, R., Li, R., Chellappa, R., 2011. Domain adaptation for object recognition: An unsupervised approach. In: Proc. Internat. Conf. on Computer Vision (ICCV), pp. 999–1006.
Arnold, A., Nallapati, R., Cohen, W., 2007. A comparative study of methods for transductive transfer learning. In: Proc. Internat. Conf. on Data Mining Workshops, pp. 77–82.
Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. Springer-Verlag, New York, Inc., New York, NY, USA.
Joachims, T., 1999. Transductive inference for text classification using support vector machines. In: Proc. Internat. Conf. on, Machine Learning (ICML), pp. 200–209.
Li, F., Wechsler, H., 2005. Open set face recognition using transduction. IEEE Trans. Pattern Anal. Machine Intell. 27 (11), 1686–1697.
Dai, W., Yang, Q., Xue, G.R., Yu, Y., 2008. Self-taught clustering. In: Proc. Internat. Conf. on, Machine Learning (ICML), pp. 200–207.
Ahonen, T., Hadid, A., Pietikainen, M., 2006. Face description with local binary patterns: Application to face recognition. IEEE Trans. Pattern Anal. Machine Intell. 28 (12), 2037–2041.
Prkachin, K.M., Solomon, P.E., 2008. The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain. PAIN 139 (2), 267–274.