Pattern Recognition Letters 37 (2014) 32–40

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Transfer learning with one-class data

Jixu Chen [a,*], Xiaoming Liu [b]

[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 11 August 2013

Keywords:
Transfer learning
Expression recognition
Landmark detection

ABSTRACT

When training and testing data are drawn from different distributions, most statistical models need to be retrained using the newly collected data. Transfer learning is a family of algorithms that improves the classifier learning in a target domain of interest by transferring the knowledge from one or multiple source domains, where the data falls in a different distribution. In this paper, we consider a new scenario of transfer learning for two-class classification, where only data samples from one of the two classes (e.g., the negative class) are available in the target domain. We introduce a regression-based one-class transfer learning algorithm to tackle this new problem. In contrast to the traditional discriminative feature selection, which seeks the best classification performance in the training data, we propose a new framework to learn the most *transferable* discriminative features suitable for our transfer learning. The experiment demonstrates improved performance in the applications of facial expression recognition and facial landmark detection.

1. Introduction

A common assumption in traditional machine learning algorithms is that the training and testing data share the same distribution. However, this assumption may not hold in many real-world applications. When the distribution changes, most statistical models need to be retrained using the newly collected data. In order to reduce the burden of recollecting and relabeling training data, the transfer learning framework is introduced (Pan and Yang, 2010; Yao and Doretto, 2010; Dai et al., 2007).

Transfer learning (TL) represents a family of algorithms that transfer the informative knowledge from a *source* domain,[1] where the training data is adequate, to a *target* domain, where the data is limited and follows a different distribution. For example, the concept of transfer learning has been explored extensively in speech recognition (Kuhn et al., 1998; Leggetter and Woodland, 1995). While the speech recognizer is trained on a large training set, its performance on a new target speaker can be poor due to the variability of human voices. On the other hand, the speeches from different speakers share many similarities. A typical TL application is speaker-adaptation, which adapts the generic speech recognition model to a new target speaker using a small amount of data collected from that speaker. Similarly, in facial expression recognition it is beneficial to adapt a generically trained expression model to a new person through TL.

In this paper, we focus on the two-class classification problem. Conventional TL algorithms assume that data samples from both positive and negative classes are available in the target domain (Chen et al., 2013). In contrast, we study a new TL setting, where only one-class data (e.g., negative data) is available in the target domain. This setting is in sharp contrast from previous TL algorithms, but is not uncommon in real-world applications. For example, in pain expression recognition, as shown in Fig. 1, a new subject has to enact the pain expression for the collection of the positive data in the target domain. This process is unnatural and cumbersome for the user, and this posed expression may be different from the spontaneous expression in the actual system execution. On the other hand, collecting the negative data (e.g., non-pain expression) of a new subject is much easier. Note that non-pain expression represents any natural expressions other than pain. The most common non-pain expression is neutral expression.

Motivated by this, we propose a regression-based algorithm to address this one-class transfer learning problem. Using the training data of one available class, we use a regressor to predict the other unknown class. Unlike the conventional imputation approach where a regressor predicts data samples, our regressor intends to predict the distributions of one class from another. The general assumption of transfer learning (Pan and Yang, 2010) is that the target and source data are different but somehow related. For example, they can share the model parameters (Yao and Doretto, 2010) or part of the training data (Dai et al., 2007; Zadrozny, 2004). In our algorithm, the basic assumption is that the

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

[1] Following the definition in Pan and Yang (2010), a domain $\mathcal{D}$ consists of two components: a feature space $\mathcal{X}$ and a marginal probability distribution $P(\mathbf{x})$, where $\mathbf{x} \in \mathcal{X}$. Here, we focus on the transfer learning when the source and target domains have different distributions.

0167-8655/$ - see front matter © 2013 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.patrec.2013.07.017

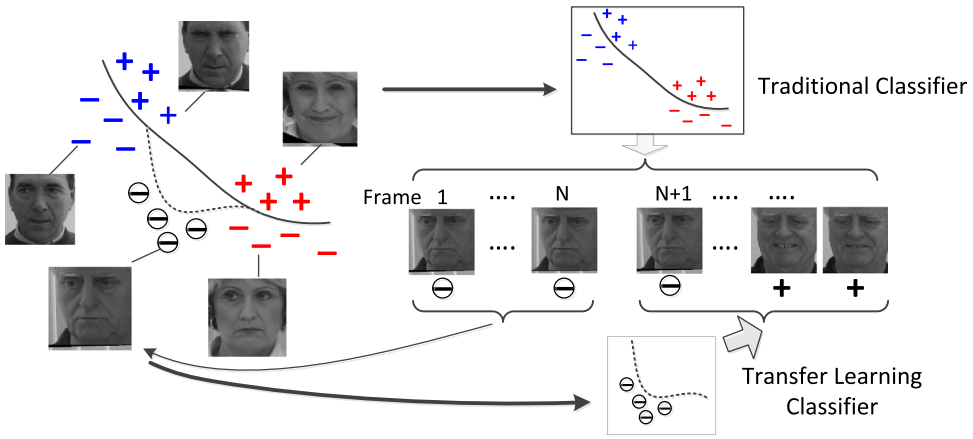

Fig. 1. Illustration of transfer learning with one-class data in pain expression recognition. Traditional classifier (solid line) is learned from the training data of different subjects and applied on a new subject. Our algorithm takes a few one-class data samples (e.g., negative samples) of the new subject and learns a new classifier. Here, + and − denote positive data samples (e.g., pain expression) and negative data samples (e.g., non-pain expression) respectively. Different colors represent different subjects. ⊖ represents the negative data in a target domain (e.g., a new subject). (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

relationship between the positive and negative distributions is shared by the target domain and the source domain.

The main contributions of this paper are as follows:

(1) We identify the novel problem of transfer learning using one-class, rather than two-class, data in the target domain. This has not been addressed before, but exists in many real-world applications.
(2) We propose a regression-based algorithm to address this problem. Because the success of our TL algorithm depends on both the classifier and the regressor, we propose a new approach to select the most *transferable features*, which are not only discriminative, but also favorable to the regressor.
(3) We design new application scenarios where the target domain performance can be improved using the readily available one-class data, such as the non-pain expression in the beginning of face videos, and the initial negative patch in facial landmark detection.

2. Related work

TL aims to extract the knowledge from one or more source domains and improve learning in the target domain. It has been applied to a wide variety of applications, such as object recognition (Yao and Doretto, 2010; Kulis et al., 2011), sign language recognition (Farhadi et al., 2007), and text classification (Wang et al., 2008).

We denote the source domain data as $\mathbf{D}_S = \{(\mathbf{x}_{S,1}, y_{S,1}), \ldots, (\mathbf{x}_{S,N_S}, y_{S,N_S})\}$ and the target domain data as $\mathbf{D}_T = \{(\mathbf{x}_{T,1}, y_{T,1}), \ldots, (\mathbf{x}_{T,N_T}, y_{T,N_T})\}$, where $\mathbf{x} \in \mathcal{X}$ is in the feature space and $y \in \{-1, +1\}$ is the binary label. Given these labels, the target and source data can be divided into positive and negative data respectively, i.e., $\mathbf{D}_S = \{\mathbf{D}_S^-, \mathbf{D}_S^+\}$ and $\mathbf{D}_T = \{\mathbf{D}_T^-, \mathbf{D}_T^+\}$.

The conventional TL algorithm can be categorized into three settings (Pan and Yang, 2010). In *inductive TL* (Dai et al., 2007; Yao and Doretto, 2010; Yang et al., 2007), both the source data $\mathbf{D}_S$ and the target data $\mathbf{D}_T$ are available. The goal is to learn the target classifier $f_T : \mathbf{x}_T \to y_T$. However, when the size of target training data $\mathbf{D}_T$ is very small, i.e., $N_T \ll N_S$, learning f_T solely from $\mathbf{D}_T$ may suffer serious overfitting problems. TL remedies this problem by using knowledge from the source data $\mathbf{D}_S$. TrAdaBoost (Dai et al., 2007) attempts to utilize the "good" source data, which is similar to the target data, to improve the target Adaboost classifier. Yao and Doretto (2010) extend TrAdaBoost to cases where abundant training data is available for multiple sources. They propose a mechanism to select the weak classifiers from the source that appears to be most closely related to the target. Kulis et al. (2011) propose a domain adaption approach for object recognition. From the labeled object categories, they learn a non-linear transformation for transferring the data points from the source to the target domain. Chen et al. (2013) propose to use inductive TL to learn a person-specific model for facial expression recognition. In this paper, we learn a person-specific model using only one-class data (negative data). Inductive TL cannot be applied in this setting directly.

In *transductive TL* (Zadrozny, 2004; Huang et al., 2006; Sugiyama et al., 2007; Si et al., 2010), the source and target data are available, but only the source data has labels. Transductive TL utilizes the unlabeled target data to "shift" or "adapt" the model in the source domain to the target domain. In the literature, transductive TL is closely related to dataset shift (Quiñonero Candela J. et al., 2008; Sugiyama et al., 2007), importance reweighting (Cortes et al., 2010; Loog, 2012; Ren et al., 2011; Zadrozny, 2004; Huang et al., 2006) and domain adaptation (Daumé and Marcu, 2006; Gopalan et al., 2011). Because the classifier f_T cannot be learned directly from the unlabeled target data, a common approach is to shift or reweight the labeled source data, from which a target classifier can be learned. Zadrozny (2004) proposes to estimate the source and target marginal distribution $P_S(\mathbf{x}_S), P_T(\mathbf{x}_T)$ independently and uses the probability ratio $\frac{P_T(\mathbf{x}_S)}{P_S(\mathbf{x}_S)}$ to reweight the source data. Huang et al. (2006) and Sugiyama et al. (2007) propose different algorithms to estimate this weight directly. The learning bound of this importance weighting approach is analyzed by Cortes et al. (2010). In the computer vision community, 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 TL is to incorporate the unlabeled target data of the source domain into the training. Si et al. (2010) propose to use the unlabeled target data as a regularization term in the discriminative subspace learning, so that the learned subspace can generalize to the target domain.

Finally, the *unsupervised TL*, such as Dai et al. (2008), is applied to a unsupervised learning task, such as clustering or dimensionality reduction, when both the target label and the source label are not available.

This paper studies a new setting of TL, where only one-class data, $\mathbf{D}_T^-$ or $\mathbf{D}_T^+$, is available in the target domain, but two-class data,

$\mathbf{D}_S^-$ and $\mathbf{D}_S^+$, are available in the source domain. To the best of our knowledge, this one-class TL problem has not been addressed in the literature. It is related to but different from the following topics:

- In transductive TL the target data is unlabeled but includes both positive and negative data, whereas in one-class TL the target data is extremely unbalanced, i.e., either positive or negative data is available.
- Similarly, semi-supervised learning utilizes a small amount of labeled data and a large amount of unlabeled data, which includes both positive and negative data.
- One-class SVM (Schölkopf et al., 2001) focuses on a one-to-many classification problem where we only have the training data of one target class. One-class SVM attempts to learn a tight hyper-sphere to include most target examples. In our one-class transfer learning, we focus on a binary classification problem in the target domain. Although only one-class data is available in the target domain, both classes are available in the source domain. Furthermore, unlike TL, one-class SVM does not consider the difference between the source and target domains.
- Similar to one-class SVM, PU-learning (Liu et al., 2003) or partially supervised learning (Liu et al., 2002) only has one-class (positive) labeled data. However, it also needs a large set of unlabeled data, from which the reliable negative samples can be selected and utilized in learning.

3. One-class transfer learning

Typically TL algorithms start with a base classifier learned from the source domain data $\mathbf{D}_S$.[2] This base classifier is then updated to a target classifier with target data. For example, in Dai et al. (2007) and Yao and Doretto (2010) the target boosted classifier is adapted from weak classifiers learned from the source data. In Duan et al. (2010) and Yang et al. (2007), the target classifier is adapted from existing SVMs from source data. The Adaboost classifier has been very popular in the vision community due to its simplicity and power to generalize well. For these reasons, we choose the Adaboost classifier (Bishop, 2006) as our base classifier.

3.1. Learning the base classifier from source data

Adaboost produces a strong classifier by combining multiple weak classifiers, such as trees or simple stumps (Friedman et al., 2000). Considering that a weak classifier will be updated with the distribution of the target data, we designed a specific form of weak classifier, which solely depends on the distribution of the positive and negative data. Here, a input data vector $\mathbf{x} = (x_1, x_2, \ldots, x_F)^T$ is composed of F features, and we model the distribution of each feature as a Gaussian distribution. The probability density function (PDF) of the fth feature x_f is

$$p_f(\mathbf{x}) = p(x_f; \mu_f, \sigma_f) = \frac{1}{\sigma_f\sqrt{2\pi}} \exp\left\{-\frac{(x_f - \mu_f)^2}{2\sigma_f^2}\right\}. \quad (1)$$

The weak classifier of the fth feature is shown in Eq. 2 of Algorithm 1.

In the source domain, since data from two classes are available, we can directly learn the Adaboost classifier as shown in Algorithm 1. Please note that each weak classifier is associated with a feature. Hence, a byproduct of the classifier learning is a set of the most discriminative feature with minimal error $\{f^{(k)}\}_{k=1..K}$.

[2] Some transductive TL algorithms (Zadrozny, 2004; Daumé and Marcu, 2006) focus on the transfer of marginal distribution $p(\mathbf{x})$. These algorithms are based on generative models without using a base classifier.

Algorithm 1. Adaboost classifier learning from the source data.

input: Source data $\mathbf{D}_S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x} \in \mathbb{R}^F$ and $y \in \{-1, +1\}$.
output: The classifier $y = H(\mathbf{x})$.
Initialize the weights $w_1^{(1)}, \ldots, w_N^{(1)} = \frac{1}{N}$.
for $k = 1$ to K **do** K is the number of weak classifiers.
  for $f = 1$ to F **do** F is the number of features.
    Estimate the distributions of positive and negative classes for the fth feature $\{p_f^+, p_f^-\}$[3]
    Specify the weak classifier as:

$$h_f(\mathbf{x}) = sign\left[log\frac{p_f^+(\mathbf{x})}{p_f^-(\mathbf{x})}\right]. \quad (2)$$

    Compute the weighted error: $\varepsilon_f^{(k)} = \sum_{i=1}^N w_i^{(k)} I(h_f(\mathbf{x}_i) \neq y_i)$, where $I(h_f(\mathbf{x}_i) \neq y_i)$ is the indicator function, which equals 1 when $h_f(\mathbf{x}_i) \neq y_i$ and 0 otherwise.
  end for
  Find the most discriminative feature with the minimal error: $f^{(k)} = \arg\min_f \varepsilon_f^{(k)}$.
  Set $\alpha^{(k)} = \frac{1}{2}\ln\left[(1 - \varepsilon_{f^{(k)}}^{(k)})/\varepsilon_{f^{(k)}}^{(k)}\right]$.
  Update the weights: $w_i^{(k+1)} = w_i^{(k)} \exp\left\{\alpha^{(k)} I\left(h_{f^{(k)}}(\mathbf{x}_i) \neq y_i\right)\right\}$.
end for
return $H(\mathbf{x}) = sign\left[\sum_k \alpha^{(k)} h_{f^{(k)}}(\mathbf{x})\right]$.

As the first attempt to address the one-class TL problem, we use a uni-modal Gaussian for simplicity, and use the feature tuning approach in Section 4.1.1 to convert a multi-modal distribution to an uni-modal distribution. This method works well in our experiments. More complex data can be approximated by a mixture of Gaussians but with the cost of increased model complexity.

3.2. One-class transfer learning from target data

Algorithm 2. One-Class (Negative) Data Transfer Learning

input: Data of M sources $\mathbf{D}_1, \ldots, \mathbf{D}_M$, where $\mathbf{D}_m = \{(\mathbf{x}_{m,1}, y_{m,1}), \ldots, (\mathbf{x}_{m,N_m}, y_{m,N_m})\}$.
The target negative data $\mathbf{D}_T^-$. The discriminative features and their weights $\{f^{(k)}, \alpha^{(k)}\}_{k=1..K}$.
output: The classifier for the target domain $y = H_T(\mathbf{x})$.
for $m = 1$ to M **do**
  for $k = 1$ to K **do**
    Estimate the distributions $\{p_{m,f^{(k)}}^-, p_{m,f^{(k)}}^+\}$ of the feature $f^{(k)}$ using $\mathbf{D}_m$.
  end for
end for
Given the distributions of M sources, learn the regressors:

[3] Because the PDF of a Gaussian distribution is determined by its parameters μ and σ, we use its parameters to denote this PDF for simplicity. For instance, the PDFs of positive and negative data distributions of the fth feature are denoted as $p_f^+ = (\mu_f^+, \sigma_f^+)$ and $p_f^- = (\mu_f^-, \sigma_f^-)$.

Algorithm 2. One-Class (Negative) Data Transfer Learning

$p^+_{f^{(k)}} = R_{f^{(k)}}(p^-_{f^{(k)}})$.
for $k = 1$ to K **do**
Estimate the negative distribution $p^-_{T_{f^{(k)}}}$ from $\mathbf{D}^-_T$.
Predict the positive distribution $\hat{p}^+_{T_{f^{(k)}}} = R^{(k)}_f(p^-_{T_{f^{(k)}}})$.
Specify the weak classifier as: $h_{T_{f^{(k)}}}(\mathbf{x}) = sign\left[log\frac{\hat{p}^+_{T_{f^{(k)}}}(\mathbf{x})}{p^-_{T_{f^{(k)}}}(\mathbf{x})}\right]$.
end for
return $H_T(\mathbf{x}) = sign\left[\sum_k \alpha^{(k)} h_{T_{f^{(k)}}}(\mathbf{x})\right]$.

In the above section, we learn the AdaBoost classifier from the positive and negative distributions of the source domain data. This classifier consists of the selected discriminative features and their weights $\{f^{(k)}, \alpha^{(k)}\}$ and the distributions of these features $\{p^+_{f^{(k)}}, p^-_{f^{(k)}}\}$. In the TL setting interested to us, the main objective is to update the base classifier given the one-class data from the target domain. One intuitive approach to achieve this objective is to only update the distributions of the selected features based on the target data, while maintaining the feature selection and their associated weights. Since only one-class target data is available, we employ a regressor to predict the distribution of the other class in the target domain. In order to learn this regressor, we assume that the source data can be divided into multiple sources $\mathbf{D}_S = \{\mathbf{D}_1, \ldots, \mathbf{D}_M\}$, e.g., the training data of facial expression recognition is from multiple subjects.

Algorithm 3 summarizes the regression-based method to update the model with only negative data $\mathbf{D}^-_T$ in the target domain. The transfer learning with positive data is the same by switching the label. Fig. 2 depicts the diagram of this one-class transfer learning.

Algorithm 3 is composed of two steps. The first step estimates the positive and negative distributions $\{p^-_m, p^+_m\}_{m=1..M}$ of M sources, which are then used as the training data to learn the regressor R between the positive and negative distributions,[4] $\hat{p}^+ = R(p^-)$, with a Gaussian Process Regression (GPR) (Rasmussen and Williams, 2005).

GPR is a non-parametric regressor and has proven its effectiveness in a wide range of applications, such as gaze estimation (Sugano et al., 2010) and object categorization (Kapoor et al., 2007). Here, we assume a noisy observation model $p^+_m = g(p^-_m) + \epsilon_m$, where each p^+_m is a function of $g(p^-_m)$ perturbed by a noise term $\epsilon_m = \mathcal{N}(0, \sigma^2)$. We set the noise variance σ^2 as the variance of p^+ in the training data, and $g(p^-_m)$ is assumed to be a Gaussian process with a covariance function:

$$k(p^-_m, p^-_l) = exp(-\|p^-_m - p^-_l\|^2). \quad (3)$$

With this assumption, given the training data $\{p^-_m, p^+_m\}_{m=1..M}$ and a new p^-, the distribution of p^+ can be derived as a Gaussian distribution, and we use its mean as the regression output:

$$\hat{p}^+ = R(p^-) = \hat{\mathbf{k}}^T(\mathbf{K} + \mathbf{S})^{-1}\mathbf{g}, \quad (4)$$

where $\mathbf{K}$ and $\mathbf{S}$ are $M \times M$ matrices whose entries are $k(p^-_m, p^-_l)$ and $\sigma^2\delta_{ml}$ respectively, and $\hat{\mathbf{k}}$ and $\mathbf{g}$ are M-dimensional vectors whose entries are $k(p^-, p^-_m)$ and p^+_m respectively. Here m and l are both matrix indexes from 1 to M. In the training of this non-parametric GPR, we only need to estimate the covariance matrix $\mathbf{K}$ from the training data.

In the second step, we estimate the distribution p^-_T from the negative target data, and predict the positive distribution $\hat{p}^+_T$ based on Eq. (4). Finally, the weak classifiers are updated using p^-_T and $\hat{p}^+_T$.

Notice that we are still using the discriminative features learned from the training data (Algorithm 1), and only update the distributions of selected features. In the next section, we will take one step further and update this feature set with the novel transferable features.

3.3. Learning the transferable features

Algorithm 3. Boosting the transferable features.

input: Data of M sources $\mathbf{D}_1, \ldots, \mathbf{D}_M$, where $\mathbf{D}_m = \{(\mathbf{x}_{m,1}, y_{m,1}), \ldots, (\mathbf{x}_{m,N_m}, y_{m,N_m})\}$.
output: Transferable features and their weights $\{\alpha^{(k)}, f^{(k)}\}_{k=1..K}$.
Step 1. Predict the positive distributions of M sources
for $f = 1$ to F **do**
for $m = 1$ to M **do**
Estimate the distributions $\{p^-_{m,f}, p^+_{m,f}\}$ of the fth feature using $\mathbf{D}_m$.
end for
for $m = 1$ to M **do**
Learn a regressor $p^+ = R_{m,f}(p^-)$ using other sources $\{p^-_{l,f}, p^+_{l,f}\}_{l=1,..,m-1,m+1,..,M}$.
Predict the positive distribution $\hat{p}^+_{m,f} = R_{m,f}(p^-_{m,f})$.
end for
end for
Step 2. Boosting transferable discriminative features
Initialize the weights $\mathbf{w}^{(1)}_1, \ldots, \mathbf{w}^{(1)}_M$ of M sources, where $\mathbf{w}_m = (w_{m,1}, \ldots, w_{m,N_m})^T$.
for $k = 1$ to K **do**
for $f = 1$ to F **do**
Specify the weak classifier as: $h_{m,f}(\mathbf{x}) = sign\left[log\frac{\hat{p}^+_{m,f}(\mathbf{x})}{\hat{p}^-_{m,f}(\mathbf{x})}\right]$.
Compute the weighted error:
$\varepsilon^{(k)}_f = \sum_{m=1}^{M}\sum_{i=1}^{N_m} w^{(k)}_{m,i} I(h_{m,f}(\mathbf{x}_{m,i}) \neq y_{m,i})$.
end for
Find the feature $f^{(k)}$ that minimizes the weighted error:
$f^{(k)} = \arg\min_f \varepsilon^{(k)}_f$.
Set $\alpha^{(k)} = \frac{1}{2}\ln\left[(1 - \varepsilon^{(k)}_f)/\varepsilon^{(k)}_f\right]$.
Update the weights:
$w^{(k+1)}_{m,i} = w^{(k)}_{m,i}\exp\{\alpha^{(k)} I(h_{m,f}(\mathbf{x}_{m,i}) \neq y_{m,i})\}$.
end for
return $\{\alpha^{(k)}, f^{(k)}\}_{k=1..K}$.

In Algorithm 1, the discriminative features are selected based on the positive and negative distributions of the source data. In contrast, for Algorithm 3, the positive distribution of the target domain is predicted through a set of regressors. Since the true positive distribution of the target data may be different from the predicted one, these features can be less than optimal for the classification task in the target domain. Hence, to remedy this issue, we

[4] We estimate one regressor for each feature f. Subscript f is ignored for simplicity. Because the variance estimation is not robust given the limited number of the target data, we only learn the regressor from the negative mean to the positive mean: $\mu^+ = R(\mu^-)$. We assume the variances of the target and source data are the same, which are estimated from all the source data.

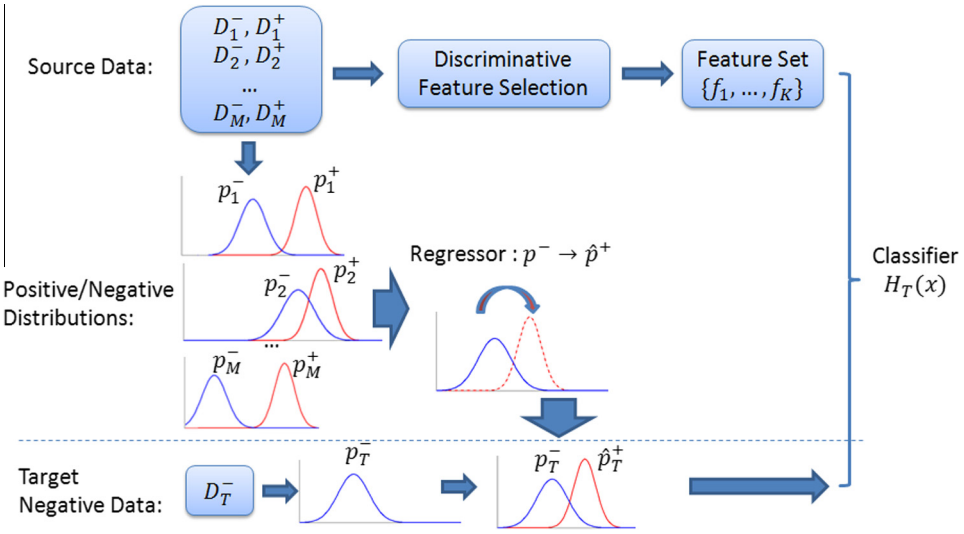

Fig. 2. Diagram of one-class transfer learning.

propose a new algorithm to select the *transferable* features which are especially designed for the one-class transfer learning setting (Algorithm 3).

Algorithm 3 consists of two steps. In the first step, for each source domain, we estimate the negative distribution from data and predict the positive distribution using a regressor trained from other $M-1$ source domains. This leave-one-source-out regressor actually simulates the regression step to be performed in the target domain during transfer learning. We repeat it for M sources to obtain the negative distributions and the predicted positive distributions $\{p_m^-, \hat{p}_m^+\}_{m=1..M}$.

The second step is similar to the discriminative feature selection in Algorithm 1. However, we use the predicted positive distribution $\hat{p}_m^+$, rather than the true positive distribution p_m^+, to learn the weak classifier. Compared to the discriminative feature, this step is consistent with our negative transfer learning which updates the weak classifiers based on the predicted positive distribution. Thus, the selected features are expected to be more suitable for the transfer learning task.

Please note that after Algorithm 3 outputs the selected transferrable features and their weights $\{\alpha^{(k)}, f^{(k)}\}_{k=1..K}$, we use Algorithm 3 to train the target model. The whole transfer learning procedure with transferable features is shown in Fig. 3. Comparing Fig. 3 with Fig. 2, the discriminative features are replaced with the transferable features.

4. Experiments

In this section, we demonstrate the efficacy of our transfer learning algorithms in two applications: pain expression recognition and facial landmark detection.

4.1. Pain expression recognition

Previous approaches (Cohen et al., 2003; Valstar et al., 2011) have shown that a person-specific model significantly outperforms a generic model when adequate person-specific data are available.

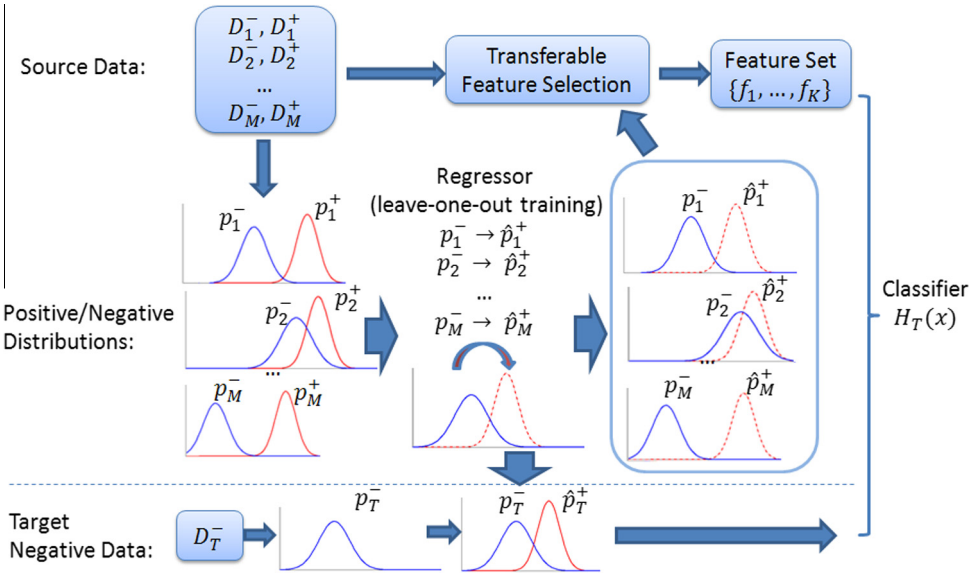

Fig. 3. Diagram of one-class transfer learning with transferable feature selection.

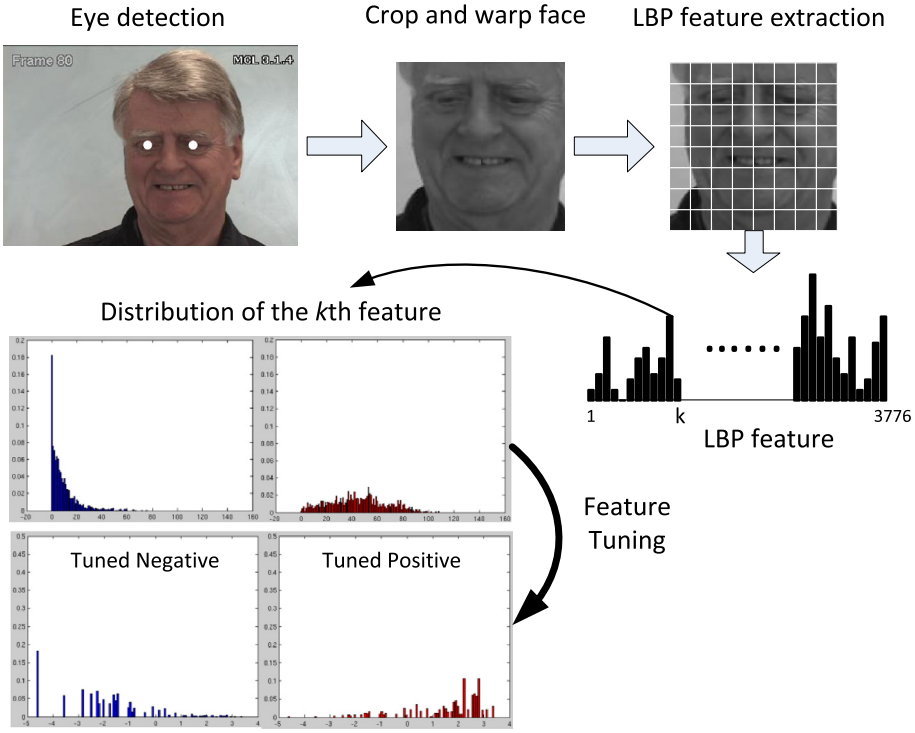

Fig. 4. LBP feature extraction and feature tuning.

However, for pain recognition, person-specific positive data is difficult to collect unless some severe conditions induce pain.

Our one-class transfer learning only needs a few negative samples to train the target model. We use the UNBC-McMaster Shoulder Pain Expression Archive database (Lucey et al., 2011a) for experiments. This database contains the spontaneous pain expression of 25 subjects with shoulder injuries during their shoulder movement. It includes 203 video sequences (totally 48,398 frames). Each frame is labeled with a pain intensity (PI) from 0 to 16. The frames with PI > 0 are labeled as positive data, and the rest frames are labeled as negative data.

4.1.1. Feature extraction

Local Binary Pattern (LBP) is used as the facial image feature because of its efficiency and effectiveness in facial expression recognition (Shan et al., 2009). Following the method in Ahonen et al. (2006), we first use the eye locations to crop and warp the face region to a 128×128 image. This face image is divided into 8×8 blocks. For each block, we extract a LBP histogram with 59 bins (please refer to Ahonen et al. (2006) for details). Finally, the LBP histograms from image blocks are concatenated into a spatially enhanced LBP feature with $59 \times 8 \times 8 = 3776$ dimensions.

Notice that our weak classifier design assumes the uni-modal Gaussian distribution of the positive and negative data. In order to handle the multi-modal distribution in real-world data, we apply the feature "tuning" approach (Collins et al., 2005) to the LBP features. This tuning step maps the feature value to the likelihood ratio of positive versus negative: $L(x) = \log \frac{max\{p^+(x),\delta\}}{max\{p^-(x),\delta\}}$, where x is the original feature value, $L(x)$ is the tuned feature value, $p^+(x)$ and $p^-(x)$ are the positive and negative distributions, and δ is a small value. We tune each dimension of the LBP feature independently. After feature tuning, the positive and negative data follow two separable uni-modal distributions, as shown in Fig. 4.

4.1.2. Pain recognition results

Similar to Lucey et al. (2011a), we perform a leave-one-subject-out cross evaluation on 25 subjects, i.e., iteratively taking one subject as the target data for testing and the remaining 24 subjects as the source data for training.

For our one-class transfer learning, the testing process on the target subject is shown in Fig. 1. To simulate the real-world application, we test on each video sequence separately. For each sequence, we assume the first few frames are non-pain expression and use those frames as negative target data. This assumption works well in real-world expression recognition systems because in real life people exhibit non-pain expression most of the time, and usually have non-pain expression when starting to use the system. In the pain database, there are 194 out of 203 sequences starting with at least 15 neutral frames. In our test, we divide each sequence into two halves. The negative frames from the first half

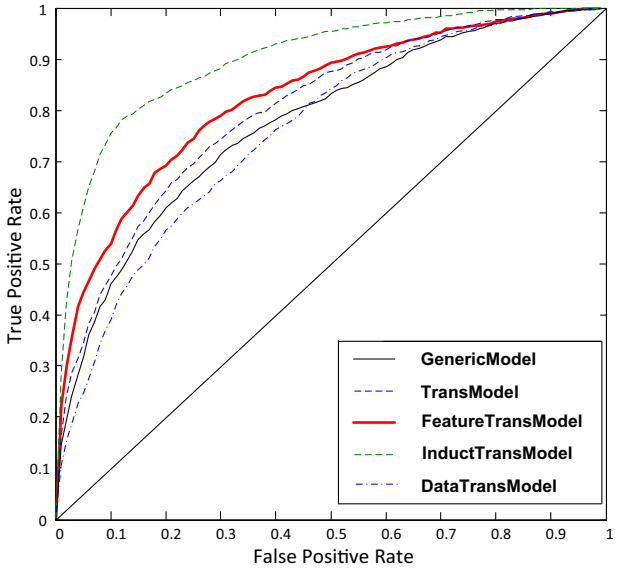

Fig. 5. ROCs of investigated algorithms.

Table 1
AUC for one-class transfer learning using different number of negative data samples N.

N	5	10	15	Quarter	Half
TransModel	0.756 ± 0.007	0.776 ± 0.006	0.783 ± 0.006	0.786 ± 0.006	0.795 ± 0.006
FeatureTransModel	0.773 ± 0.007	0.799 ± 0.006	0.802 ± 0.006	0.804 ± 0.006	0.820 ± 0.006
Generic Model	0.771 ± 0.007				

are used for transfer learning, and we test on the second half. The number of frames for each sequence varies from 66 to 683, with an average of 238.

The first experiment compares the performance of five algorithms:

- *GenericModel* is a baseline approach using the source data to train a generic Adaboost classifier (Algorithm 1);
- *TransModel* is learned using the transfer learning algorithm as described in Algorithm 3. It shares the same feature set as *GenericModel*, but its positive and negative distributions are updated with the target data;
- *FeatureTransModel* uses Algorithm 3 to select the transferable features, and Algorithm 3 to update the distributions;
- *InductTransModel* is an state-of-the-art inductive transfer learning algorithm (Yao and Doretto, 2010) utilizing both positive and negative data in the target domain. We use the parameter transfer learning as described in Yao and Doretto (2010).
- *DataTransModel* is a naive data transfer learning algorithm that directly combines the negative target data with all positive source data and learn an Adaboost classifier. This algorithm uses the same data as *TransModel* and *FeatureTransModel* use.

All five classifiers use the same number (400) of weak classifiers. The ROC curves of the above algorithms are shown in Fig. 5. The area under ROC curves (AUC) is 0.771 for *GenericModel*, 0.795 for *TransModel*, 0.820 for *FeatureTransModel*, 0.895 for *InductTransModel*, and 0.753 for *DataTransModel*.

First, we notice that *DataTransModel* is even worse than the generic model, because we only update the negative data distribution using the negative target data, without considering the transfer of the positive data distribution. Second, we can see that the one-class transfer learning can improve the generic model. With the discriminative feature, transfer learning improves the baseline slightly, but with the selected transferable features, the AUC is improved significantly from 0.771 to 0.82. The state-of-the-art pain recognition system (Lucey et al., 2011b) achieved AUC=0.751 using appearance features, and achieved AUC = 0.839 by combining two difference normalized appearance features and a shape feature. Compared to Lucey et al. (2011b), our algorithm needs neutral expression for transfer learning, and can achieve comparable result by only using appearance features.

Note that it is not a fair comparison between *InductTransModel* and one-class transfer learning, because the former uses both positive and negative target data. Hence, it may be viewed as a loose upper-bound of our methods. There are two inductive transfer learning algorithms, i.e., instance-transfer and parameter-transfer in Yao and Doretto (2010). Both of them select weak classifiers based on the error rate in the target data. If only negative target data is available, they tend to select weak classifiers to classify all the data as negative, so that the classification error on training data is zero, but the error on the testing data is very large. When $N = half$, the AUC is 0.54 for *InductTransModel* with negative target data only.

Although the negative data samples are usually easy to collect, we would like to use as fewer data samples as possible in practical applications. To evaluate the effect of the transfer learning data size, we select the first N negative frames from the testing sequence $N = 5, 10, 15, quarter, half$. For *quarter* and *half*, we use all the negative frames from the first quarter or the half of the testing sequence for transfer learning. Table 1 shows the AUC[5] of the transfer learning algorithms using different numbers of negative data samples N. Compared to *GenericModel*, *FeatureTransModel* can improve the AUC from 0.771 to 0.802 with the first 15 negative frames, which is less than 1 s of the video clip.

Our transfer learning is very efficient, since it only needs to compute the mean of the target data. It runs in real time (< 30 ms) on a PC with 3.2 GHz CPU. For algorithm training, the transferable feature selection is time consuming (~ 25 min), but it is only slightly slower than the generic model training (~ 21.5 min).

4.1.3. Comparison of two feature sets

As we discussed in Section 3.3, transferable feature selection is optimal for our regression-based transfer learning. To demonstrate the efficacy of transferable features, we compare them with the discriminative features (Algorithm 1) regarding their classification and regression accuracy. The top $1, 5, 10, 20, 50, 100, 200$ and 400 features selected by two different methods are compared. To evaluate the regression accuracy we compare the predicted positive mean ($\hat{\mu}^+$) and the true mean (μ^+) of the positive data.[6] The average regression error for top N features is: $\frac{1}{N}\sum_{i=1}^{N}|\hat{\mu}_i^+ - \mu_i^+|$. The regression and classification errors are shown in Fig. 6.

We observe that the top transferable features are better than discriminative features in both classification and regression tasks. We also observe that the top selected features have larger regression errors than the features selected later. That is because a feature is selected based on its classification ability. The top features tend to have both larger data variances and larger distances between positive and negative data. The large regression error may be due to the large data variance. Considering the variance of the positive and negative data, we use the metric ΔD to measure the regression performance: $\Delta D = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|\hat{\mu}_i^+ - \mu_i^+|}{\sigma_i^+} - \frac{|\hat{\mu}_i^+ - \mu_i^-|}{\sigma_i^-}\right)$, where $\mu_i^{+,-}$ and $\sigma_i^{+,-}$ are the mean and standard deviation of the positive and negative distribution respectively. This metric is the average difference between the normalized distance from the positive mean and the normalized distance from the negative mean. ΔD gets smaller when $\hat{\mu}_i^+$ is closer to μ_i^+ and further from μ_i^-. As shown in Fig. 6, the top transferable features have much smaller ΔD compared to discriminative features. Please notice that ΔD is negative because $\hat{\mu}_i^+$ is always closer to μ_i^+ than to μ_i^-. This means that the predicted positive mean is not only close to the true positive mean but also far from the true negative mean. This also explains why these transferable features yield better classification results.

4.2. Facial landmark detection

Face alignment aims to estimate the location of a set of facial landmarks (e.g., eye corner, mouth corner) on a given image by

[5] The upper-bound of the uncertainty of AUC is computed by $\sigma = \sqrt{\frac{AUC(1-AUC)}{min(n^+, n^-)}}$ (Cortes and Mohri, 2004), where n^+, n^- are the number of positive and negative testing data samples.

[6] Positive data is only used for this evaluation. It is not available in the test run.

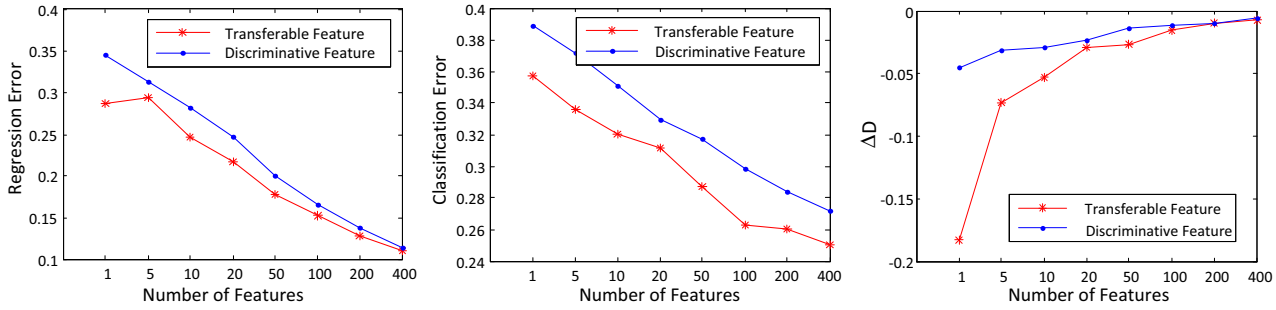

Fig. 6. Comparison of the transferable feature and the discriminative feature.

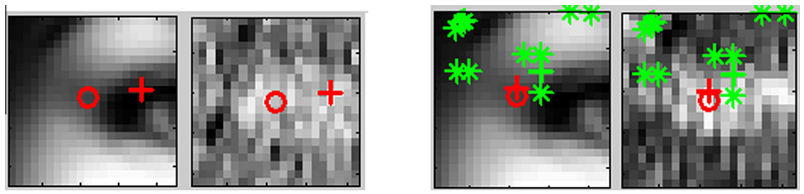

Fig. 7. Detection results for the generic LAM (a) and LAM with transfer learning (b). The red circle is the ground-truth. The red cross is the detection result. The green cross and stars represent the initial position and positions to extract negative transfer learning data. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

using a supervisely learned model (Matthews and Baker, 2004; Liu, 2009). One popular model is the Active Shape Model (Cristinacce and Cootes, 2007), which employs a set of local appearance models (LAMs) to localize each landmark independently, and then uses a global shape model to constrain the solution. Taking the eye corner as an example, during the training procedure, its LAM is discriminatively learned from the positive data (patch extracted from labeled ground-truth location) and the negative data (patch extracted from the neighboring locations). During the testing process, given an initial location, LAM will be applied to all candidate locations within a neighborhood, and the maximum classifier score will determine the estimated eye corner location. Since the LAM is critical to the alignment performance, we apply transfer learning to improve its performance. We view the generic training data as the source domain and the given test image as the target domain. Hence, the local patches around the initial location of the test image are negative samples of the target domain.

We use the Labeled Face Parts in the Wild (LFPW) database (Belhumeur et al., 2011), which includes images under a wide range of head poses, light conditions, and image qualities. A total of 35 fiducial landmarks are manually labeled in each face image. Our experiment only focuses on the LAM of the left eye corner, but the algorithm can be applied to other landmarks as well.

As the eye corner appearance varies substantially cross different images, a generic eye corner LAM may not work well for an arbitrary unseen face image. In contrast, adapting the generic LAM using the specific image characteristics embodied in the negative data might result in a better LAM for this particular test image.

We randomly split 1135 images from LFPW into 200 training images and 935 testing images. First, all the images are rotated and scaled based on the labeled eye positions. Since each image is labeled four times by different labelers, we extract four 15×15 patches as the positive samples and randomly select 7 negative positions around ground-truth to extract negative samples. Each 15×15 patch is reshaped to a 115-dimensional vector and tuned using the same feature tuning method as described in Section 4.1.1. Similarly, we can extract the positive and negative data from testing images.

First, we train a generic eye corner LAM from training data using Algorithm 1. However, this classifier performs poorly on the testing data (AUC $= 0.613 \pm 0.011$). Since this classifier works well on the training data (AUC $= 0.940 \pm 0.008$), this poor testing result is contributed by the large variation between the training and testing data. To address this problem, in each testing image, we extract 5 negative examples around the initial landmark location, and use our one-class transfer learning algorithm to update the classifier. This updated classifier improves the AUC to 0.665 ± 0.011.

To test our classifiers for eye corner detection, we start from an initial eye corner position, which is detected by the PittPatt face-eye detector,[7] and search a neighborhood around the initial position to find the maximal classifier output as our detection result. For our transfer learning, we randomly extract a few negative examples around the initial position to update the classifier. An example of the detection result and the classifier output score map of two LAMs are shown in Fig. 7. The red circle is the ground truth. The red cross is the detection result. The green cross and stars represent the initial position and positions to extract negative transfer learning data. The results show that the generic LAM fails because of the appearance of a large eye shadow. By updating the model with examples from a small neighborhood, our transfer learning can improve the classification result and more importantly, results in a score map with fewer high scores (fewer bright pixels), which indicates the improved detection reliability. In our experiment, we randomly select an initial position which is up to $0.15d$ (d is the interocular distance) from the ground truth. The average detection error over 935 testing images is $0.0919d$ for the generic LAM and $0.0834d$ for the transfer learning LAM. Although the improvement may appear to be small, it is significant ($t = 4.48, p < 0.05$ in t-test). Here, no comparison is performed between our one-landmark detection result and the state-of-the-art face alignment algorithm (e.g., Belhumeur et al., 2011), since Belhumeur et al. (2011) detects 35 landmarks jointly with the help of a global face shape model.

[7] http://www.pittpatt.com/

5. Conclusions

This work identified a new problem of transfer learning, where only one-class data is available. This problem is not uncommon in real-world applications, but has not been studied before. We introduced a new regression-based one-class transfer learning algorithm to address this problem. In this algorithm, we introduced a new feature selection framework for selecting the *transferable features* that are not only discriminative between the negative and positive data, but also excellent in predicting the positive data distribution from the negative data. We applied our algorithm to facial expression recognition and facial landmark detection. Compared to the generic model without transfer learning, our algorithm with the transferable features can improve both applications with only a few negative examples. This is the first attempt to address such a one-class transfer learning problem. Our framework is general and applicable to a wide range of learning problems where only one-class target data is available. The main assumption of our algorithm is that multiple sources are required, each consisting of a pair of positive and negative distributions. Some applications, like object recognition in Yao and Doretto (2010), aims at solving a one-to-many classification problem, where multiple sources may share the same large background. Our algorithm cannot be applied to such a setting. Another limitation is that our algorithm is only applicable to classifiers that are directly derived from data distribution, such as the specific Adaboost classifier described in Section 3.1. Further research is required in order to generalize it to other classifiers such as SVM and kNN.

References

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

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

Dai, W., Yang, Q., Xue, G.R., Yu, Y., 2007. Boosting for transfer learning. In: Proc. of the International Conference on Machine Learning (ICML), pp. 193–200.

Kuhn, R., Nguyen, P., Junqua, J.C., Goldwasser, L., Niedzielski, N., Fincke, S., et al., 1998. Eigenvoices for speaker adaptation. In: Proc. of the International Conference on Spoken Language Processing (ICSLP), pp. 1771–1774.

Leggetter, C.J., Woodland, P., 1995. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. Comput. Speech Lang. 9, 171–185.

Chen, J., Liu, X., Tu, P., Aragones, A., 2013. Learning person-specific models for facial expression and action unit recognition. Pattern Recognition Letter 34 (15), 1964–1970.

Zadrozny, B., 2004. Learning and evaluating classifiers under sample selection bias. In: Proc. of the International Conference on Machine Learning (ICML), pp. 903–910.

Kulis, B., Saenko, K., Darrell, T., 2011. What you saw is not what you get: domain adaptation using asymmetric kernel transforms. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 1785–1792.

Farhadi, A., Forsyth, D., White, R., 2007. Transfer learning in sign language. In: Proc. of the 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. of the International Conference on Data Mining (ICDM), pp. 1085–1090.

Yang, J., Yan, R., Hauptmann, A.G., 2007. Cross-domain video concept detection using adaptive SVMs. In: Proc. of the International Conference on Multimedia, pp. 188–197.

Huang, J., Smola, A.J., Gretton, A., Borgwardt, K.M., Schölkopf, B., 2006. Correcting sample selection bias by unlabeled data. In: Advances in Neural Information Processing Systems (NIPS), pp. 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. In: Advances in Neural Information Processing Systems (NIPS), pp. 1433–1440.

Si, S., Tao, D., Geng, B., 2010. Bregman divergence-based regularization for transfer subspace learning. IEEE Trans. Knowledge Data Eng. 22, 929–942.

Quiñonero Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D. (Eds.), 2008. Dataset Shift In Machine Learning. MIT Press.

Cortes, C., Mansour, Y., Mohri, M., 2010. Learning bounds for importance weighting. In: Advances in Neural Information Processing Systems (NIPS), pp. 442–450.

Loog, M., 2012. Nearest neighbor-based importance weighting. In: IEEE International Workshop on Machine Learning for, Signal Processing (MLSP), pp. 1–6.

Ren, S., Hou, Y., Zhang, P., Liang, X., 2011. Importance weighted AdaRank. In: Proceedings of the 7th International Conference on Advanced Intelligent Computing, pp. 448–455.

Daumé III, H., Marcu, D., 2006. Domain adaptation for statistical classifiers. J. Art. Intell. Res. 26 (1), 101–126.

Gopalan, R., Li, R., Chellappa, R., 2011. Domain adaptation for object recognition: an unsupervised approach. In: Proc. of the Intl. Conf. on Computer Vision (ICCV), pp. 999–1006.

Dai, W., Yang, Q., Xue, G.R., Yu, Y., 2008. Self-taught clustering. In: Proc. of the International Conference on, Machine Learning (ICML), pp. 200–207.

Schölkopf, B., Platt, J.C., Shawe-Taylor, J.C., Smola, A.J., Williamson, R.C., 2001. Estimating the support of a high-dimensional distribution. Neural Comput. 13, 1443–1471.

Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.S., 2003. Building text classifiers using positive and unlabeled examples. In: Proc. of the International Conference on Data Mining (ICDM), pp. 179–186.

Liu, B., Lee, W.S., Yu, P.S., Li, X., 2002. Partially supervised classification of text documents. In: Proc. of the International Conference on, Machine Learning (ICML), pp. 387–394.

Duan, L., Xu, D., Tsang, I., Luo, J., 2010. Visual event recognition in videos by learning from web data. In: Proc. of the IEEE Conf. on Computer Vision and, Pattern Recognition (CVPR), pp. 1959–1966.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer.

Friedman, J., Hastie, T., Tibshirani, R., 2000. Special invited paper-additive logistic regression: a statistical view of boosting. Ann. Stat. 28, 337–407.

Rasmussen, C.E., Williams, C.K.I., 2005. Gaussian Processes for Machine Learning. MIT Press.

Sugano, Y., Matsushita, Y., Sato, Y., 2010. Calibration-free gaze sensing using saliency maps. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 2667–2674.

Kapoor, A., Grauman, K., Urtasun, R., Darrell, T., 2007. Active learning with gaussian processes for object categorization. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 1–8.

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

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

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. of Int. Conf. on Automatic Face and Gesture Recognition (FG), pp. 57–64.

Shan, C., Gong, S., McOwan, P.W., 2009. Facial expression recognition based on local binary patterns: a comprehensive study. J. Image Vision Comput. 27 (6), 803–816.

Ahonen, T., Hadid, A., Pietikainen, M., 2006. Face description with local binary patterns: application to face recognition. IEEE Trans. Pattern Anal. Mach. Intell. 28 (12), 2037–2041.

Collins, R., Liu, Y., Leordeanu, M., 2005. On-line selection of discriminative tracking features. IEEE Trans. Pattern Anal. Mach. Intell. 27 (1), 1631–1643.

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

Cortes, C., Mohri, M., 2004. Confidence intervals for the area under the roc curve. In: Advances in Neural Information Processing Systems (NIPS), pp. 305–312.

Matthews, I., Baker, S., 2004. Active appearance models revisited. Int. J. Comput. Vision 60 (2), 135–164.

Liu, X., 2009. Discriminative face alignment. IEEE Trans. Pattern Anal. Mach. Intell. 31 (11), 1941–1954.

Cristinacce, D., Cootes, T., 2007. Boosted regression active shape models. In: Proc. of the British Machine Vision Conference (BMVC), vol. 2, University of Warwick, UK, pp. 880–889.

Belhumeur, P.N., Jacobs, D.W., Kriegman, D.J., Kumar, N., 2011. Localizing parts of faces using a consensus of exemplars. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 545–552.