Common Subspace Learning via Cross-Domain Extreme Learning Machine

Yan Liu[1] · Lei Zhang[1] · Pingling Deng[1] · Zheng He[1]

Received: 7 October 2016 / Accepted: 25 April 2017 / Published online: 5 May 2017

Abstract Extreme learning machine (ELM) is proposed for solving a single-layer feed-forward network (SLFN) with fast learning speed and has been confirmed to be effective and efficient for pattern classification and regression in different fields. ELM originally focuses on the supervised, semi-supervised, and unsupervised learning problems, but just in the single domain. To our best knowledge, ELM with cross-domain learning capability in subspace learning has not been exploited very well. Inspired by a cognitive-based extreme learning machine technique (Cognit Comput. 6:376–390, 1; Cognit Comput. 7:263–278, 2.), this paper proposes a unified subspace transfer framework called cross-domain extreme learning machine (CdELM), which aims at learning a common (shared) subspace across domains. Three merits of the proposed CdELM are included: (1) A cross-domain subspace shared by source and target domains is achieved based on domain adaptation; (2) ELM is well exploited in the cross-domain shared subspace learning framework, and a new perspective is brought for ELM theory in heterogeneous data analysis; (3) the proposed method is a subspace learning framework and can be combined with different classifiers in recognition phase, such as ELM, SVM, nearest neighbor, etc. Experiments on our electronic nose olfaction datasets demonstrate that the proposed CdELM method significantly outperforms other compared methods.

Keywords Extreme learning machine (ELM) · Subspace learning · Domain adaptation · Electronic nose

Introduction

Although many cognitive computational models (e.g. neural networks and support vector machines, etc.) have been proposed to solve classification problem, those methods encounter lots of challenges, such as poor computational scalability, trivial human intervention, etc. Recently, extreme learning machines (ELM) [1–5], as a cognitive-based technique, are proposed for "generalized" Single-Layer Feed-forward Networks (SLFNs) [1, 2, 4, 6, 7]. ELM can analytically determine the output layer using Moor-Penrose generalized inverse by adopting the square loss of prediction error. Huang et al. [4, 6, 8, 9] have rigorously proved that, in theory, ELM can approximate any continuous functions and also proved the classification capability of ELM [4]. Moreover, different from traditional learning algorithms [10], ELM tends to achieve not only the smallest training error but also the smallest norm of output weights for better generalization performance [3, 7]. Its variants [11–19] also focus on the regression, classification, and pattern recognition applications.

For the past years, there are a number of improved versions of ELM. A weighted ELM was proposed for binary/multiclass classification tasks with both balanced and imbalanced data distribution [20]. Bai et al. [21] proposed a sparse ELM for reducing storage space and testing time. Huang et al. [22] proposed a semi-supervised ELM for classification, in which

✉ Lei Zhang
leizhang@cqu.edu.cn

Yan Liu
yanliu@cqu.edu.cn

Pingling Deng
dengpl@cqu.edu.cn

Zheng He
ZhengHe@cqu.edu.cn

[1] College of Communication Engineering, Chongqing University,
No.174 Shazheng street, Shapingba district, Chongqing 400044,
China

a manifold regularization with graph Laplacian was imposed as constraint, and an unsupervised ELM was also explored for clustering. Liu et al. [23–25] proposed many methods to address the issue of tactile object recognition. Recently, Liu et al. [26] also proposed an extreme kernel sparse learning for tactile object recognition, which combines the ELM and kernel dictionary learning.

The data distribution obtained in different stages with different sampling conditions would be different, which is identified as cross-domain problem. Traditional ELM assumes that the data distribution between training and testing data should be similar and therefore cannot address this issue. To handle this problem, domain adaptation has been proposed for heterogeneous data (i.e. cross-domain problem), by leveraging a few labeled instances from another domain with similar semantic [27, 28]. Inspired by domain adaptation, Zhang et al. [29] proposed a domain adaption ELM (DAELM) for classification across tasks (source domain vs. target domain), which is the first paper to study the cross-domain learning mechanism in ELM. However, DAELM was proposed as a cross-domain classifier, and how to learn a shared (common) subspace with source and target domains, to our knowledge, has never been studied in ELM. Therefore, in this paper, we extend ELMs to handle cross-domain problems by transfer learning and subspace learning, and explore its capability in multi-domain application for ELMs. Inspired by DAELM, we propose a cross-domain extreme learning machine (CdELM) for common subspace learning, the basic idea of the proposed CdELM method is illustrated in Fig. 1, in which we aim at learning a shared subspace β. Notably, the proposed CdELM is different from DAELM that we would like to learn a shared subspace for source and target domains, instead of a shared classifier.

The remainder of this is as follows. Section 2 contains a brief review of ELM. In Section 3, the proposed CdELM method with detailed model formulation and optimization algorithm is presented. The experiments and results have been discussed in Section 4. Finally, Section 5 concludes this paper.

Related Work

Review of ELM

Briefly, the principle of ELM is described as follows. Given the training data $X = [x_1, x_2, \ldots, x_N] \in \Re^{N \times n}$, where n is the dimensionality and N is the number of training samples, and $T = [t_1, t_2, \ldots, t_N] \in \Re^{N \times m}$ denotes the labels with respect to the data $\boldsymbol{X}$, where m is the number of classes. The output of the hidden layer is denoted as $\mathscr{H}(x_i) \in \Re^{1 \times L}$, where L is the number of hidden nodes and $\mathscr{H}(\cdot)$ is the activation function. The output weights between the hidden layer and the output layer being learned is denoted as $\beta \in \Re^{L \times m}$. The regularized

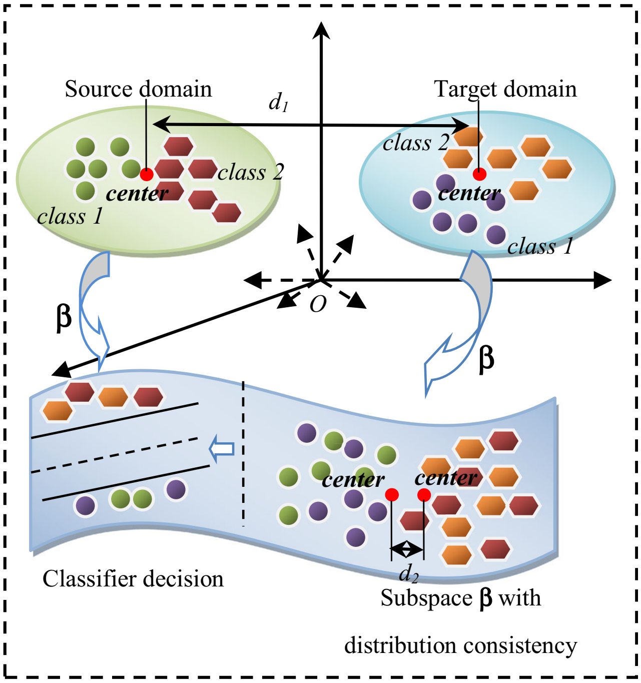

Fig. 1 Schematic diagram of the proposed CdELM method; after a subspace projection β, the source domain and target domain of different space distribution lie in a latent subspace with good distribution consistency (the centers of both domains become very close and drift is removed). Formally, the upper coordinate system denotes the raw data points of source domain and target domain in three dimensions. We use the word "center" to represent the mean of each domain data. From the upper figure, we can see that the difference between the mean of source domain and the mean of target domain is large. After a subspace projection β in the *below* figure, we can see that the values of d_2 become smaller, which demonstrate that the distribution difference becomes small, and both domains of different space distribution lie in a latent common subspace with good distribution consistency

ELM aims at minimizing the training error and the norm of the output weights for better generalization performance, formulated as follows:

$$\min_{\beta} \mathscr{L} = \frac{1}{2}\|\beta\|^2 + \frac{C}{2}\sum_{i=1}^{N}\|\xi_i\|^2 \; s.t. \mathscr{H}(x_i)\beta = t_i - \xi_i, i = 1, \ldots, N \quad (1)$$

where ξ_i denotes the prediction error with respect to the ith training pattern x_i and C is a penalty constant on the training errors.

By substituting the constraint term in (1) into the objective function, an equivalent unconstrained optimization problem can be obtained as follows:

$$\min_{\beta} \mathscr{L} = \frac{1}{2}\|\beta\|^2 + \frac{C}{2}\sum_{i=1}^{N}\|T - H\beta\|^2 \quad (2)$$

where $H = [\mathscr{H}(x_1); \mathscr{H}(x_2); \ldots; \mathscr{H}(x_N)] \in \Re^{N \times L}$.

The minimization problem (2) is a regularized least square problem. By setting the gradient of $\mathscr{L}$ with respect to β to zero, we can get the closed-form solution of β. There are two cases while solving β. If N is larger than L, the gradient equation is over-determined, and the closed-form solution can be obtained as

$$\beta^* = \left(H^T T + \frac{I}{C}\right)^{-1} H^T T \tag{3}$$

where $\boldsymbol{I}$ denotes the identity matrix of size L.

Second, if the number N of training samples is smaller than L, an under-determined least square problem would be handled. In this case, the solution of (1) can be obtained as

$$\beta^* = H^T\left(HH^T + \frac{I}{C}\right)^{-1} T \tag{4}$$

where $\boldsymbol{I}$ denotes the identity matrix of size N.

Therefore, for classification problem, the solution of β can be computed by using Eq. (3) or Eq. (4). We direct the interested readers to [3] for more details on ELM theory and the algorithms.

Subspace Learning

Subspace learning aims at learning a low-dimensional subspace. There are several common methods, such as principal component analysis (PCA) [30], linear discriminant analysis (LDA) [31], and manifold learning-based locality preserving projections (LPP) [32]. All of these methods suppose that the data distribution is consistent, namely, that they are only applicable to single domain. However, this assumption is often violated in many real-world applications. So, for heterogeneous data, we proposed a new cross-domain learning method to learn a shared subspace, which is called cross-domain extreme learning machine.

The Proposed CdELM Method

Notations

In this paper, source domain and target domain are defined by subscript S and T, respectively. The training data of source and target domain is denoted as $X_S = \left[x_S^1, \ldots, x_S^{N_S}\right] \in \mathfrak{R}^{D\times N_S}$ and $X_T = \left[x_T^1, \ldots, x_T^{N_T}\right] \in \mathfrak{R}^{D\times N_T}$, respectively, where D is the number of dimensions, N_S and N_T are the number of training samples in both domains. Let $\beta \in \mathfrak{R}^{L\times d}$ represent the basis transformation that maps the ELM space of source and target data to some subspace with dimension of d. $\|\cdot\|_F$ and $\|\cdot\|_2$ denote the Frobenius norm and l_2-norm. $Tr(\cdot)$ denotes the trace operator and $(\cdot)^T$ denotes the transpose operator. Throughout this paper, matrix is written in capital bold face, vector is presented in lower bold face, and variable is in italics.

The Proposed Method

As illustrated in Fig. 1, the distribution between the source domain and target domain is different. Therefore, the performance of the learned classifier by the source domain will be dramatically degraded. Inspired by subspace learning and ELM, the main idea of the proposed CdELM is to learn a shared subspace β in ELM space rather than a classifier β. Therefore, the source domain and target domain share the similar feature distribution in the latent projection β.

Firstly, mapping the source data and target data into the ELM space, and then we could obtain $H_S = \left[h_S^1, \ldots, h_S^{N_S}\right] \in \mathfrak{R}^{L\times N_S}$ and $H_T = \left[h_T^1, \ldots, h_T^{N_T}\right] \in \mathfrak{R}^{L\times N_T}$, where $h_S^i = g\left(W^T x_S^i + b^T\right)$ and $h_T^j = g\left(W^T x_T^j + b^T\right)$ are the output (column) vector of the hidden layer with respect to the input x_S^i and x_T^j, respectively, i = 1,2,...,N_S, j = 1,2,...,N_T, $g(\cdot)$ is a activation function, L is the number of randomly generated hidden nodes, $W \in \mathfrak{R}^{D\times L}$ and $b \in \mathfrak{R}^{1\times L}$ are randomly generated weights.

In the learned subspace β, we expect that not only the distribution between source domain and target domain should be consistent, and the discrimination can also be improved for recognition. Inspired by linear discriminant analysis (LDA), we aim at minimizing the intra-class scatter matrix S_W^S and simultaneously maximizing the inter-class scatter matrix S_B^S of the source data, such that the separability can be promised in the learned linear subspace. Therefore, for source domain, it is rational to maximize the following term

$$\max_{\beta} \frac{Tr\left(\beta^T S_B^S \beta\right)}{Tr\left(\beta^T S_W^S \beta\right)} \tag{5}$$

where the inter-class scatter matrix and intra-class scatter matrix can be computed as $S_B^S = \sum_{c=1}^{C} \left(\mu_S^c - \mu_S\right)\left(\mu_S^c - \mu_S\right)^T$ and $S_W^S = \sum_{c=1}^{C} \sum_{k=1, h_S^k \in G_c}^{n_c} \left(h_S^k - \mu_S^c\right)\left(h_S^k - \mu_S^c\right)^T$, where μ_S represents the center of source data, μ_S^c represents the center of class c of source data in the raw space, C represents the number of categories, G_c represents a collection belonging to class c, and n_c represents the number of class c.

For learning such a subspace β that maximizes the formulation (5), we should also ensure that the projection does not distort the data from target domain, such that much more available information can be kept in the new subspace representation. Therefore, it is rational to maximize the following term:

$$\max_{\beta} Tr\left(\left(\beta^T H_T\right)\left(\beta^T H_T\right)^T\right) = \max_{\beta} Tr\left(\beta^T H_T H_T^T \beta\right) \tag{6}$$

Table 1 Data description of the E-nose data

E-nose System	Formaldehyde	Benzene	Toluene	Carbon monoxide	Nitrogen dioxide	Ammonia	Total
Master	126	72	66	58	38	60	420
Slave 1	108	108	106	98	107	81	608
Slave 2	108	108	94	95	108	84	576

Naturally, after projected by β, the feature distributions between the mapped source domain $H_S = \left[h_S^1, \ldots, h_S^{N_S}\right] \in \mathfrak{R}^{L\times N_S}$ and target domain $H_T = \left[h_T^1, \ldots, h_T^{N_T}\right] \in\mathfrak{R}^{L\times N_T}$ can become similar. Therefore, it is rational to have an idea that the mean distribution discrepancy (MDD) between H_S and H_T can be minimized. That is, the distance between the centers of the two domains should be minimized. Therefore, the MDD minimization is formulated as

$$\min \left\| \frac{1}{N_S}\sum_{i=1}^{N_S} \beta^T h_S^i - \frac{1}{N_T}\sum_{j=1}^{N_T} \beta^T h_T^j \right\|_F^2 \tag{7}$$

With the merits of ELM, we expect that the norm of β is minimized,

$$\min_{\beta} \|\beta\|_F^2 \tag{8}$$

After a detailed description of the four specific parts in the proposed CdELM model, by incorporating the Eq. (5) to Eq. (8) together, a complete CdELM model is formulated as follows

$$\min_{\beta} \frac{Tr\left(\beta^T S_W^S \beta\right) + \lambda_0\|\beta\|_F^2 + \lambda_1 \left\| \frac{1}{N_S}\sum_{i=1}^{N_S} \beta^T h_S^i - \frac{1}{N_T}\sum_{j=1}^{N_T} \beta^T h_T^j \right\|_F^2}{Tr\left(\beta^T S_B^S \beta\right) + \lambda_2 Tr\left(\beta^T H_T H_T^T \beta\right)} \tag{9}$$

where λ_0, λ_1, and λ_2 denote the trade-off parameters.

Let $\mu_S = \frac{1}{N_S}\sum_{i=1}^{N_S} h_S^i$ and $\mu_T = \frac{1}{N_T}\sum_{j=1}^{N_T} h_T^j$ be the centers of source domain and target domain in ELM space; then the minimization problem in Eq.(9) can be finally written as

$$\begin{aligned}
&\min_{\beta} \frac{Tr\left(\beta^T S_W^S \beta\right) + \lambda_0\|\beta\|_F^2 + \lambda_1 \left\| \beta^T \left(\frac{1}{N_S}\sum_{i=1}^{N_S} h_S^i\right) - \beta^T \left(\frac{1}{N_T}\sum_{j=1}^{N_T} h_T^j\right) \right\|_F^2}{Tr\left(\beta^T S_B^S \beta\right) + \lambda_2 Tr\left(\beta^T H_T H_T^T \beta\right)} \\
&= \min_{\beta} \frac{Tr\left(\beta^T S_W^S \beta\right) + \lambda_0\|\beta\|_F^2 + \lambda_1 \left\|\beta^T \mu_S - \beta^T \mu_T\right\|_F^2}{Tr\left(\beta^T S_B^S \beta\right) + \lambda_2 Tr\left(\beta^T H_T H_T^T \beta\right)} \\
&= \min_{\beta} \frac{Tr\left(\beta^T S_W^S \beta\right) + \lambda_0 Tr\left(\beta^T \beta\right) + \lambda_1 Tr\left(\left(\beta^T \mu_S - \beta^T \mu_T\right)\left(\beta^T \mu_S - \beta^T \mu_T\right)^T\right)}{Tr\left(\beta^T S_B^S \beta + \lambda_2 \beta^T H_T H_T^T \beta\right)} \\
&= \min_{\beta} \frac{Tr\left(\beta^T S_W^S \beta + \lambda_0 \beta^T \beta + \lambda_1 \left(\beta^T \mu_S - \beta^T \mu_T\right)\left(\beta^T \mu_S - \beta^T \mu_T\right)^T\right)}{Tr\left(\beta^T S_B^S \beta + \lambda_2 \beta^T H_T H_T^T \beta\right)} \\
&= \min_{\beta} \frac{Tr\left(\beta^T \left(S_W^S + \lambda_0 I + \lambda_1 (\mu_S - \mu_T)(\mu_S - \mu_T)^T\right)\beta\right)}{Tr\left(\beta^T \left(S_B^S + \lambda_2 H_T H_T^T\right)\beta\right.}
\end{aligned} \tag{10}$$

where $\boldsymbol{I}$ is an identity matrix of size L.

Model Optimization

In the minimization problem Eq. (10), there are many possible solutions of β (i.e. non-unique solutions). To guarantee the unique property of solution, we impose an equality constraint on the optimization problem, and then Eq. (10) can be written as

$$\begin{aligned}
&\min_{\beta} Tr\left(\beta^T \left(S_W^S + \lambda_0 I + \lambda_1 (\mu_S - \mu_T)(\mu_S - \mu_T)^T\right)\beta\right) \\
&s.t. Tr\left(\beta^T \left(S_B^S + \lambda_2 H_T H_T^T\right)\beta\right) = \eta
\end{aligned} \tag{11}$$

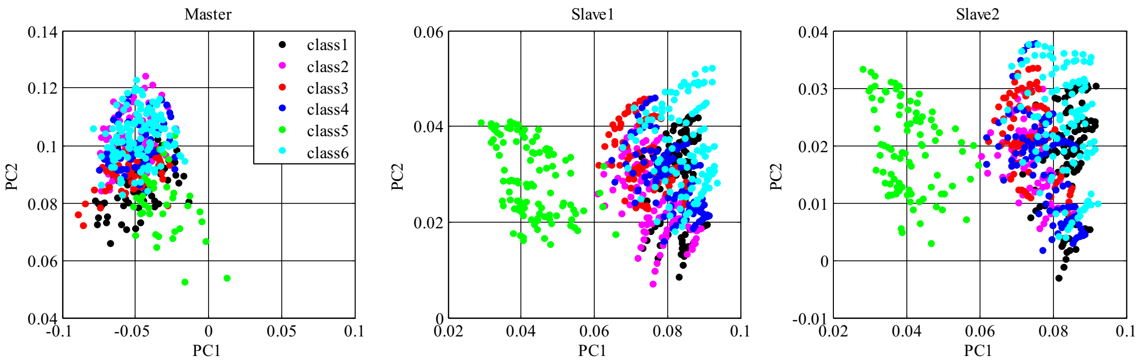

Fig. 2 The PCA scatter points of the master, slave 1, and slave 2 data, respectively

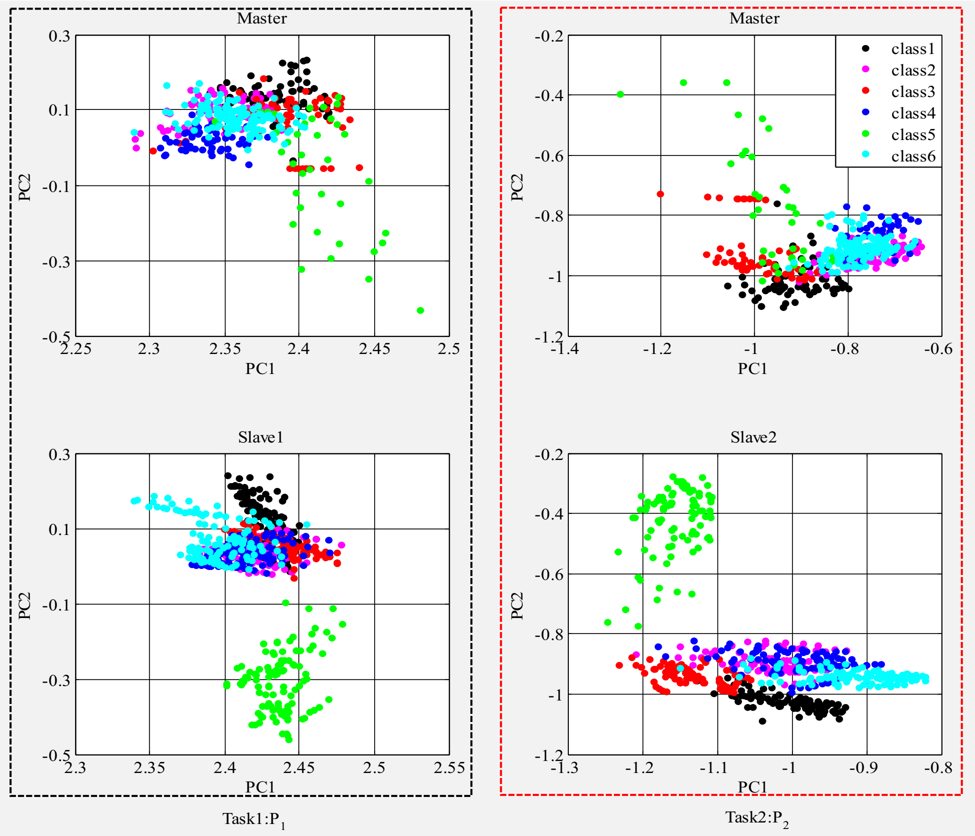

Fig. 3 The scatter points by using CdELM from master → slave 1 (*black dot*) and master → slave 2 (*red dot*)

where η is a positive constant.

To solve Eq. (11), the Lagrange multiplier function is written as

$$\mathscr{L}(\beta,\rho)=\beta^T\left(S_W^S+\lambda_0 I+\lambda_1(\mu_S-\mu_T)(\mu_S-\mu_T)^T\right)\beta-\rho\left(\beta^T\left(S_B^S+\lambda_2 H_T H_T^T\right)\beta-\eta\right) \qquad (12)$$

where ρ denotes the Lagrange multiplier coefficient.

By setting the partial derivation of $\mathscr{L}(\beta,\rho)$ with respect to β to be 0, we have

$$\frac{\partial\mathscr{L}(\beta,\rho)}{\partial\beta}=0\rightarrow\left(S_B^S+\lambda_2 H_T H_T^T\right)^{-1}\left(S_W^S+\lambda_0 I+\lambda_1(\mu_S-\mu_T)(\mu_S-\mu_T)^T\right)\beta=\rho\beta \qquad (13)$$

From Eq. (13), we can observe that β can be obtained by solving the following eigenvalue decomposition problem,

$$A\beta=\rho\beta \qquad (14)$$

where $A=\left(S_B^S+\lambda_2 H_T H_T^T\right)^{-1}\left(S_W^S+\lambda_0 I+\lambda_1(\mu_S-\mu_T)(\mu_S-\mu_T)^{\ T}\right)$ and ρ denotes the eigenvalues.

From (14), it is clear that β denotes the eigenvectors. Due to that the model (11) is a minimization problem; therefore, the optimal subspace denotes the eigenvectors with respect to the first d smallest eigenvalues $[\rho_1,..., \rho_d]$, represented by

$$\beta^*=[\beta_1,\beta_2,\ldots,\beta_d] \qquad (15)$$

For easy implementation, the proposed algorithm is summarized in Algorithm 1.

Table 2 Recognition accuracy (%) with sensor calibration under setting 1

Cross-domain tasks	SVM	ELM (sigmoid)	ELM (rbf)	KernelELM	PCA	LDA	LPP	NPE	NCA	MDS	LFDA	SGF	CdELM (sigmoid)	CdELM (rbf)
Master → slave 1	51.97	55	54.59	53.63	55.05	55.56	53.95	53.62	41.28	51.15	61.84	55.10	64.90	**66.39**
Master → slave 2	60.59	59.83	61.16	61.93	60.88	61.09	57.81	54.69	33.85	58.51	61.63	57.49	68.11	**68.45**

The bold entries indicate the best results

Table 3 Recognition accuracy (%) with sensor calibration under setting 2 (task 1)

Methods	n_t					Average
	1	3	5	7	9	
SVM	59.14	63.22	62.80	70.49	70.76	65.28
ELM(sigmoid)	59.05	65.10	69.52	71.20	72.18	67.41
ELM(rbf)	62.46	65.83	67.61	69.36	69.87	67.03
PCA	63.92	67.32	70.02	73.36	73.83	69.69
LDA	63.84	67.83	70.33	71.48	73.45	69.39
LPP	65.46	69.83	71.45	72.08	71.48	70.06
NPE	64.78	64.07	63.49	71.55	71.84	67.15
NCA	52.49	50.85	53.81	50.00	63.00	54.03
MDS	61.13	64.75	65.57	70.32	72.92	66.94
LFDA	62.13	67.12	71.63	76.86	74.91	70.53
SGF	66.61	66.95	67.13	70.14	72.74	68.71
CdELM(sigmoid)	67.88	71.53	73.50	**75.34**	76.36	72.92
CdELM(rbf)	**68.34**	**73.32**	**74.42**	75.07	**77.35**	**73.70**

The bold entries indicate the best results

Table 4 Recognition accuracy (%) with sensor calibration under setting 2 (task 2)

Methods	n_t					Average
	1	3	5	7	9	
SVM	69.65	72.76	73.63	74.16	74.90	73.02
ELM(sigmoid)	64.18	67.35	67.91	68.09	69.29	67.36
ELM(rbf)	65.35	68.41	68.63	68.71	69.60	68.14
PCA	65.84	69.27	70.18	72.23	72.15	69.93
LDA	65.23	68.76	69.85	71.74	73.07	69.73
LPP	69.82	74.19	73.63	72.85	76.82	73.46
NPE	71.05	71.15	71.43	72.28	74.14	72.01
NCA	56.32	47.49	52.01	55.62	58.03	53.90
MDS	72.11	73.48	73.81	75.28	75.29	74.00
LFDA	65.26	70.61	73.08	75.47	77.01	72.29
SGF	68.07	71.51	73.08	73.03	73.56	71.85
CdELM(sigmoid)	**74.47**	75.35	75.78	77.56	78.09	76.25
CdELM(rbf)	74.13	**76.06**	**77.69**	**78.95**	**79.18**	**77.21**

The bold entries indicate the best results

Algorithm 1.

```
Input: Heterogeneous data including data $\mathbf{X}_S \in \Re^{D\times N_S}$ and data $\mathbf{X}_T \in \Re^{D\times N_T}$, hidden nodes $L$, $\lambda_0$, $\lambda_1$, $\lambda_2$ and $d$
Procedure:
    1. Randomly generated weights $\mathbf{W} \in \Re^{D\times L}$, $\mathbf{b} \in \Re^{1\times L}$, and choose the activation function $g(\cdot)$
    2. Compute $\mathbf{H}_S = [\mathbf{h}_S^1,\ldots,\mathbf{h}_S^{N_S}]$ and $\mathbf{H}_T = [\mathbf{h}_T^1,\ldots,\mathbf{h}_T^{N_T}]$, where $\mathbf{h}_S^i = g(\mathbf{W}^{\mathrm{T}}\mathbf{x}_S^i + \mathbf{b}^{\mathrm{T}})$ and $\mathbf{h}_T^j = g(\mathbf{W}^{\mathrm{T}}\mathbf{x}_T^j + \mathbf{b}^{\mathrm{T}})$
    3. Compute the inter-class scatter matrix $\mathbf{S}_B^S$ and the intra-class scatter matrix $\mathbf{S}_W^S$
    4. Compute the matrix A as $(\mathbf{S}_B^S + \lambda_2\mathbf{H}_T\mathbf{H}_T^{\mathrm{T}})^{-1}(\mathbf{S}_W^S + \lambda_0\mathbf{I} + \lambda_1(\boldsymbol{\mu}_S - \boldsymbol{\mu}_T)(\boldsymbol{\mu}_S - \boldsymbol{\mu}_T)^{\mathrm{T}})$
    5. Perform the eigenvalue decomposition of A by using (14)
    6. Compute the optimal subspace $\boldsymbol{\beta}^* = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_d]$ by (15)
Output: The basis transformation $\boldsymbol{\beta}$
```

Experiments

Data Description

We validate the proposed method on our own datasets. This dataset includes three subsets: master data (collected 5 years ago), slave 1 data (collected now), and slave 2 data (collected now). In data acquisition, the master and two slavery E-nose systems were developed in [33]. Each system consists of four TGS series sensors and an extra temperature and humidity module. Therefore, the dimension of each sample is 6. This dataset includes six kinds of gaseous contaminants (i.e. six classes), such as formaldehyde, benzene, toluene, carbon monoxide, nitrogen dioxide, and ammonia. The detailed description of the dataset is shown in Table 1. For visually observing the heterogeneous E-nose data, the PCA scatter points on the master data, slave 1 data and slave 2 data, are shown in Fig. 2, in which we can see that the points from different classes are overlapped.

Experimental Settings

The master data collected 5 years ago is used as source domain data (no drift). The slave 1 and slave 2 data collected now are used as target domain data (with drift). Then, we conduct the experiments with two settings as follows.

Setting 1. During CdELM training and classifier learning, the labels of the target domain data are unavailable, and only the source labels are used. The classification accuracy on slave 1 data (or slave 2 data) is reported.

Setting 2. The only difference between *Setting 1* and *Setting 2* is that, in classifier training, partial labeled data of target domain can be used. Specifically, for each class in the target domain, k labeled samples can be used for classifier learning, where the values $k = 1$, 3, 5, 7, and 9 are discussed in this paper.

Compared Methods

To show the effectiveness of the proposed method, we have chosen 12 machine learning methods. First, three baseline

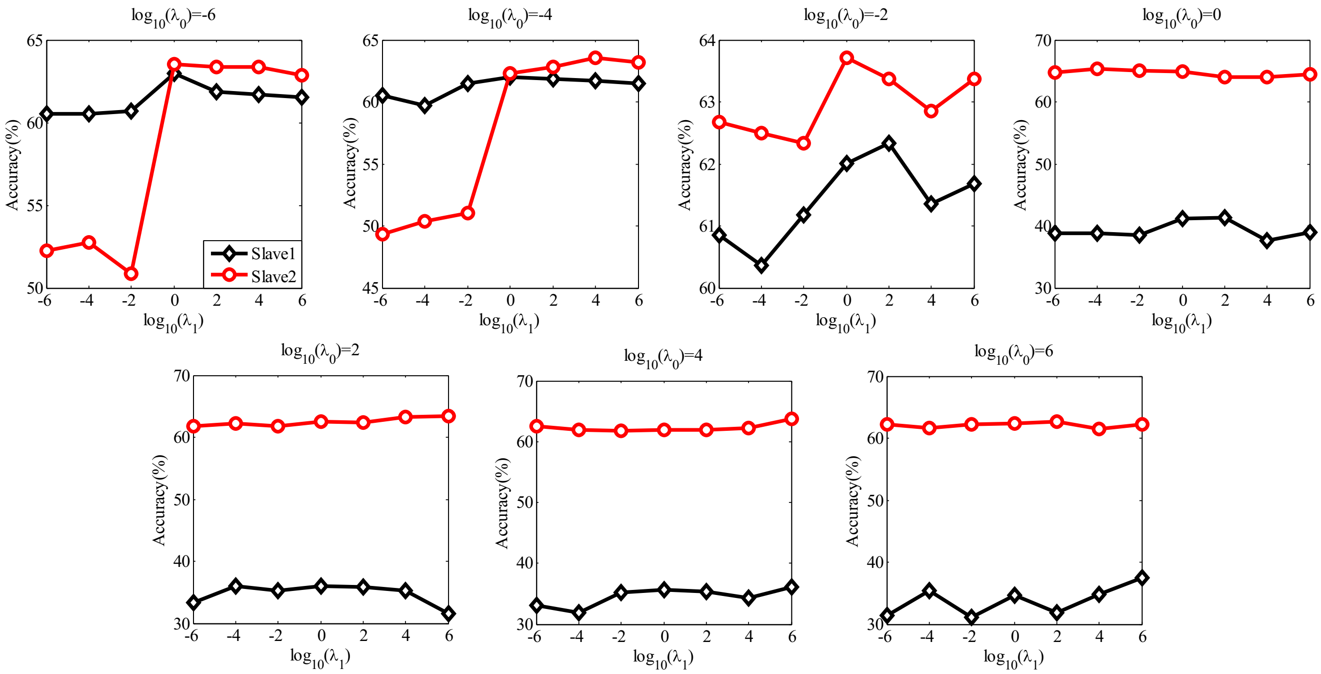

Fig. 4 The performance curves with respect to λ_1 under different values of λ_0

methods such as support vector machine (SVM), principal component analysis (PCA), and linear discriminant analysis (LDA) are compared. Second, five semi-supervised learning methods based manifold learning, including locality preservation projection (LPP) [32], multidimensional scaling (MDS) [34], neighborhood component analysis (NCA) [35], neighborhood preserving embedding (NPE) [36], and local fisher discriminant analysis (LFDA) [37] are explored and compared. Finally, the popular subspace transfer learning method, sampling geodesic flow (SGF) [38], is also explored and compared.

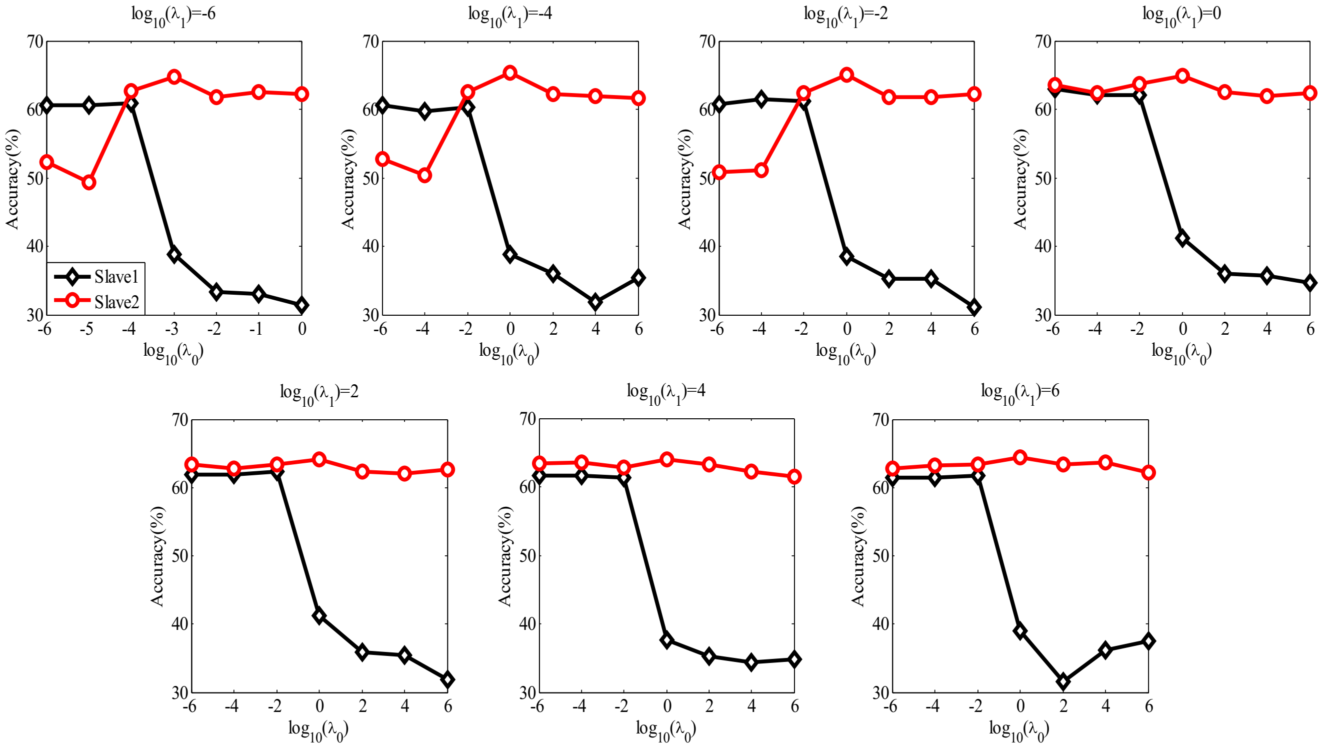

Fig. 5 The performance curves with respect to λ_0 under different values of λ_1

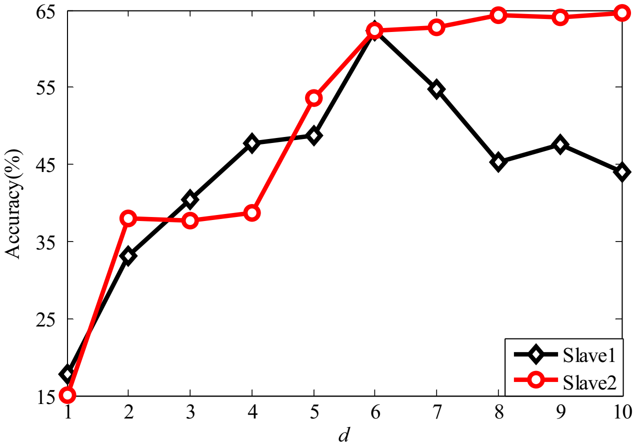

Fig. 6 The performance curves with respect to subspace dimensionality *d*

Results

In this section, the experimental results on each setting are reported to validate the performance of the proposed CdELM method. Under each setting, two tasks including master → slave 1 and master → slave 2 are conducted.

Under setting 1, we first observe the qualitative result by implementing the proposed CdELM method on master → slave 1 and master → slave 2, respectively. The result is shown in Fig. 3, in which the separability among data points from different classes (represented as different symbols) is much improved in the learned common subspace compared to Fig. 2. Further, the odor classification accuracy of the target domain data has been presented in Table 2. From the results, we can observe that the proposed CdELM achieves the highest accuracies on two tasks. While the activation function in hidden layer is Gaussian (RBF function), the best performance of the CdELM is achieved. This demonstrates that the proposed CdELM has a good performance for cross-domain pattern recognition scenarios.

Under the setting 2, *k*-labeled data for each class in the target domain is leveraged for classifier learning. The recognition accuracy of the first task (i.e. master → slave 1) is reported in Table 3 and the second task (i.e. master → slave 2) is reported in Table 4. Notably, all the compared methods follow the same setting conditions. From Tables 3 and 4, we can observe that the proposed CdELM still outperforms other methods. Therefore, we confirm that the proposed method is effective in handling heterogeneous measurement data.

Parameter Sensitivity Analysis

In the proposed CdELM model, there are three parameters: λ_0, λ_1, and d. We focus on observing the performance variations in tuning λ_0 and λ_1 according to 10^t, where $t = \{-6, -4, -2, 0, 2, 4, 6\}$. To show the performance with respect to each parameter, one is tuned by freezing the other one. The parameter λ_1 tuning results by fixing λ_0 are shown in Fig. 4, and the parameter λ_0 tuning results by fixing λ_1 are shown in Fig. 5, from which the best parameters λ_0 and λ_1 can be witnessed. Further, we tune the subspace dimensionality d from the parameter set $d = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, and the result is shown in Fig. 6 by fixing other model parameters.

Conclusion

In this paper, we present a cross-domain common subspace learning approach for heterogeneous data classification problem, which is called cross-domain extreme learning machine (CdELM). The method is motivated by subspace learning, domain adaptation, and cognitive-based extreme learning machine, such that the advantage of ELM, such as good generalization, is inherited. Since traditional ELM supposes that the training data and testing data should be with similar distribution, once the assumption is violated in multi-domain scenarios, the ELM may not be adapted. The aim of this paper is to bring some new perspective for ELM in multi-domain subspace learning scenarios. Extensive experiments demonstrate that the proposed method outperforms other compared methods.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grant 61401048, the Fundamental Research Funds for the Central Universities, and Chongqing University Postgraduates' Innovation Project (No.CYB15030).

Compliance with Ethical Standards

Conflict of Interest Yan Liu, Lei Zhang, Pingling Deng, and Zheng He declare that they have no conflict of interest.

Informed Consent Informed consent was not required as no human or animals were involved.

Human and Animal Rights This article does not contain any studies with human or animal subjects performed by any of the authors.

References

1. Huang GB. An insight into extreme learning machines: random neurons, random features and kernels. Cognit Comput. 2014;6(3):376–90.
2. Huang GB. What are extreme learning machines? Filling the gap between frank Rosenblatt's dream and John von Neumann's puzzle. Cognit Comput. 2015;7(3):263–78.
3. Huang GB, Zhu QY, Siew CK. Extreme learning machine: theory and applications. Neurocomputing. 2006;70(1–3):489–501.
4. Huang GB, Zhou H, Ding X, Zhang R. Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern B Cybern. 2012;42(2):513–29.

5. Cambria E. Extreme learning machines [trends & controversies]. IEEE Intell Syst. 2013;28(6):30–59.
6. Huang GB, Chen L, Siew CK. Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw. 2006;17(4):879–92.
7. Huang GB, Zhu QY, Siew CK. Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proc. IEEE Int Joint Conf Neural Netw. vol. 2. pp. 985–990, 2004.
8. Huang GB, Chen L. Convex incremental extreme learning machine. Neurocomputing. 2007;70(16–18):3056–62.
9. Huang GB, Chen L. Enhanced random search based incremental extreme learning machine. Neurocomputing. 2008;71(16–18): 3460–8.
10. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagation errors. Nature. 1986;323:533–6.
11. Huang GB, Zhu QY, Mao KZ, Siew CK, Saratchandran P. Can threshold networks be trained directly? IEEE Trans Circ Syst II Exp Briefs. 2006;53(3):187–91.
12. Liang NY, Huang GB, Saratchandran P, Sundararajan N. A fast and accurate on-line sequential learning algorithm for feedforward networks. IEEE Trans Neural Netw. 2006;17(6):1411–23.
13. Li MB, Huang GB, Saratchandran P, Sundararajan N. Fully complex extreme learning machine. Neurocomputing. 2005;68:306–14.
14. Feng G, Huang GB, Lin Q, Gay R. Error minimized extreme learning machine with growth of hidden nodes and incremental learning. IEEE Trans Neural Netw. 2009;20(8):1352–7.
15. Rong HJ, Huang GB, Sundararajan N, Saratchandran P. Online sequential fuzzy extreme learning machine for function approximation and classi fication problems. IEEE Trans Syst Man Cybern B Cybern. 2009;39(4):1067–72.
16. Huang Z, Yu Y, Gu J and Liu H. An efficient method for traffic sign recognition based on extreme learning machine. IEEE Trans Cybern, Appear Online, 2016.
17. Yu Y, Sun Z. Sparse coding extreme learning machine for classification. Neurocomputing, accepted, 2016.
18. Zhang L, Zhang D. Evolutionary cost-sensitive extreme learning machine. IEEE Trans Neural Netw Learn Syst. 2016; doi:10.1109/TNNLS.2016.2607757.
19. Zhang L, Zhang D. Robust Visual knowledge transfer via extreme learning machine based domain. IEEE Trans Image Process. 2016;25(10):4959–73.
20. Zong W, Huang GB, Chen Y. Weighted extreme learning machine for imbalance learning. Neurocomputing. 2013;101:229–42.
21. Bai Z, Huang GB, Wang D, Wang H, Westover MB. Sparse extreme learning machine for classification. IEEE Trans Cybern. 2014;44(10):1858–70.
22. Huang GB, Song S, Gupta JND, Wu C. Semi-supervised and unsupervised extreme learning machines. IEEE Trans. Cybern. 2014; doi:10.1109/TCYB.2014.2307349.
23. Liu H, Yu Y, Sun F, et al. Visual-tactile fusion for object recognition. IEEE Trans Autom Sci Eng. 2016;14:1–13.
24. Liu H, Guo D, Sun F. Object recognition using tactile measurements: kernel sparse coding methods. IEEE Trans Instrum Meas. 2016;65(3):1–10.
25. Liu H, Liu Y, Sun F. Robust exemplar extraction using structured sparse coding. IEEE Trans Neural Netw Learn Syst. 2014;26(8): 1816–21.
26. Liu H, Qin J, Sun F, et al. Extreme kernel sparse learning for tactile object recognition. IEEE Trans Cyber, 2016.
27. Pan SJ, Tsang IW, Kwok JT, Yang Q. Domain adaptation via transfer component analysis. IEEE Trans Neural Netw. 2011;22(2):199–210.
28. Zhang L, Zuo W, Zhang D. LSDT: latent sparse domain transfer learning for Visual adaptation. IEEE Trans Image Process. 2016;25(3):1177–91.
29. Zhang L, Zhang D. Domain adaptation extreme learning machines for drift compensation in e-nose systems. IEEE Trans Instrum Meas. 2015;64(7):1790–801.
30. Turk M, Pentland A. Eigenfaces for recognition. J Cogn NeuroSci. 1991;3(1):71–86.
31. Belhumeur PN, Hespanha J, Kriegman DJ. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell. 1997;19(7):711–20.
32. He X, Niyogi P. Locality preserving projections. NIPS, 2004.
33. Zhang L, Tian F. Performance study of multilayer Perceptrons in a low-cost electronic nose. IEEE Trans Instrum Meas. 2014;63(7): 1670–9.
34. Torgerson WS. Multidimensional scaling: I. Theory and method. Psychometrika. 1952;17(4):401–19.
35. Goldberger J, Roweis S, Hinton G, Salakhutdinov R. Neighborhood component analysis. NIPS, 2004.
36. He X, Cai D, Yan S, and Zhang HJ. Neighborhood preserving embedding. ICCV, 2005.
37. Sugiyama M. Local fisher discriminant analysis for supervised dimensionality reduction. ICML, pp. 905–912, 2006.
38. Gopalan R, Li R, and Chellappa R. Domain adaptation for object recognition: an unsupervised approach. ICCV, 2011.