Domain Adaptation Extreme Learning Machines

Lei Zhang

[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong

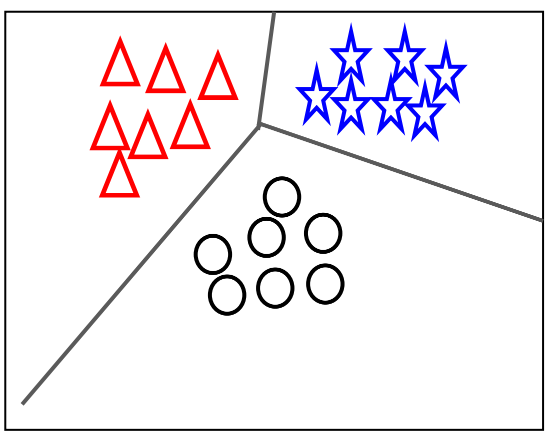
[2]College of Communication Engineering, Chongqing University, Chongqing, China

2014-12-8

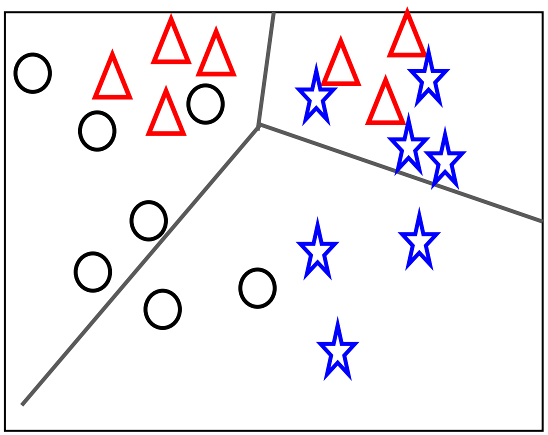
The 5th International Conference on Extreme Learning Mchines, Singapore, Dec 8-10,2014

Motivation

- **Original Problem:** **Can one classifier learned on a training dataset from source domain fit the data in target domain that has different feature space distribution?**



(a) Source domain

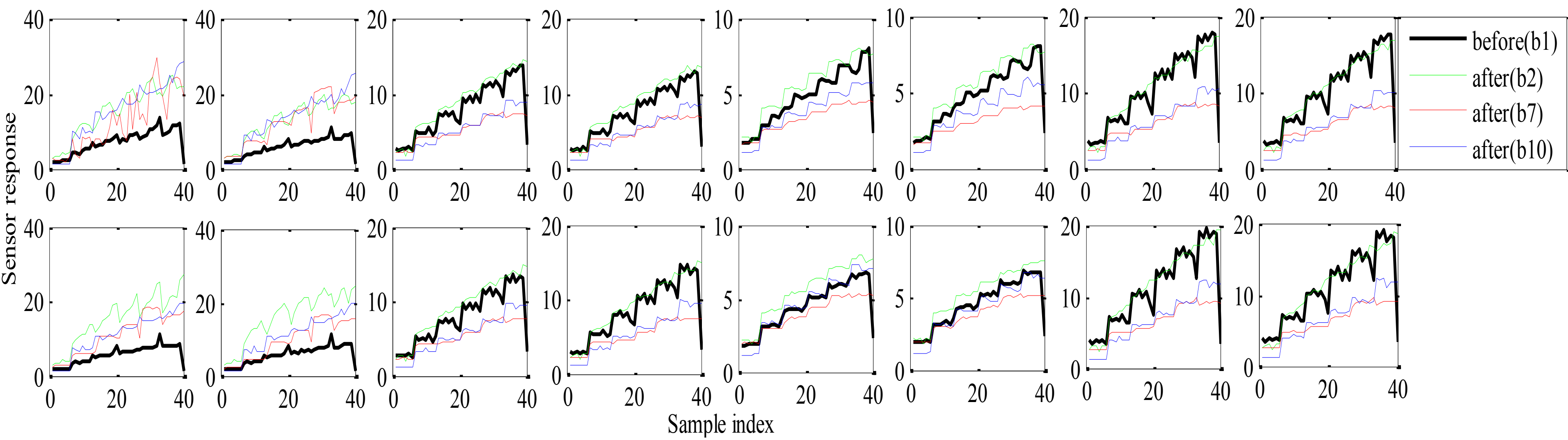


(b) Target domain, no adaptation

Motivation

Real-world applications:

- **Sub-problem 1: In machine olfaction, large scale sensory data of odor perception is from different domains due to the inherent sensor drift cause by aging, poisoning, ad-hoc experimental environment.**

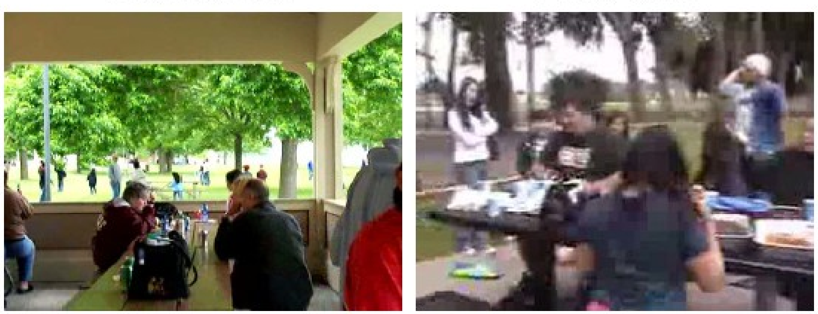


[1] L. Zhang, D. Zhang, "Domain Adaptation Extreme Learning Machines for Drift Compensation in E-Nose Systems," IEEE Transactions on Instrumentation and Measurement, 2014, in press.

Motivation

Real-world applications:

- **Sub-problem 2: In computer vision, visual event recognition in consumer videos is a challenging due to two main issues:**
  - **consumer videos are generally captured by amateurs using hand-held cameras.**
  - **annotate many consumer videos is reluctant, and result in weakly generalized classifier learned from a limited number of labeled training videos.**



(a) "picnic"

(b) "sports"

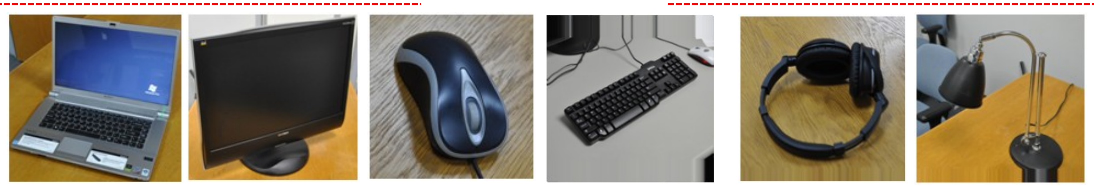
[2] L. Zhang, D. Zhang, "Robust Visual Knowledge Transfer vis EDA", IEEE Transactions Pattern Analysis and Machine Intelligence, Submitted, 2014.

Motivation

Real-world applications:

- **Sub-problem 3:** **In object recognition, due to the ad-hoc environment, different feature space and statistical properties (e.g. mean, intraclass, interclass variance) are caused among datasets?**

(a) amazon.com



(b) Digital SLR camera

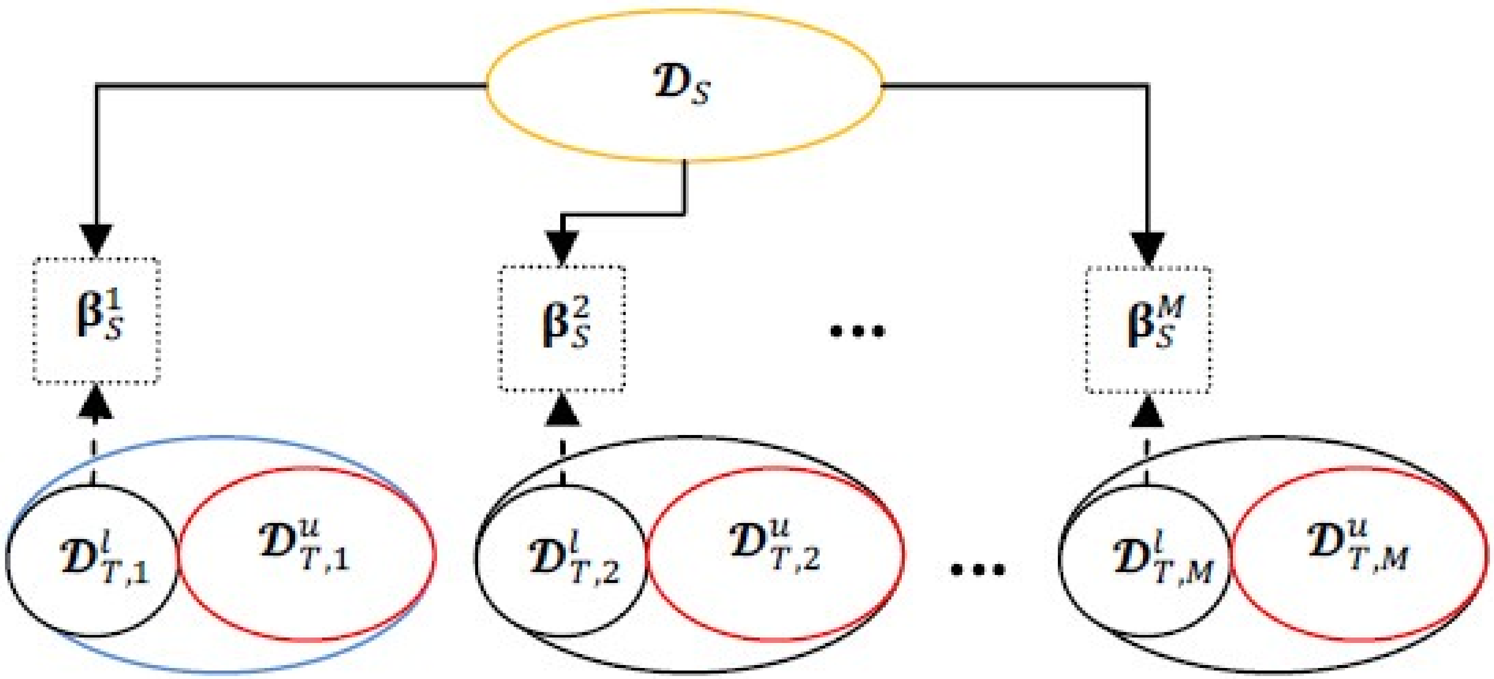
(c) robot-mounted webcam

In this talk

- The Proposed Domain Adaptation ELM: Models and Algorithms
- The Electronic Nose and Sensor Drift Understanding
- Experiments and Results
- Future work in Computer Vision

Source domain based: DAELM-S

- The proposed DAELM-S aims to learn a classifier using all labeled instances from the source domain by leveraging a limited number of labeled data from target domain.



Source domain based: DAELM-S

Formulation of DAELM-S

$$\min_{\boldsymbol{\beta}_S,\boldsymbol{\xi}_S^i,\boldsymbol{\xi}_T^i} \frac{1}{2}\|\boldsymbol{\beta}_S\|^2 + C_S\frac{1}{2}\sum_{i=1}^{N_S}\left\|\boldsymbol{\xi}_S^i\right\|^2 + C_T\frac{1}{2}\sum_{j=1}^{N_T}\left\|\boldsymbol{\xi}_T^j\right\|^2$$

$$\text{s.t.}\begin{cases}\mathbf{H}_S^i\boldsymbol{\beta}_S = \mathbf{t}_S^i - \boldsymbol{\xi}_S^i, i = 1, \dots, N_S \\ \mathbf{H}_T^j\boldsymbol{\beta}_S = \mathbf{t}_T^j - \boldsymbol{\xi}_T^j, j = 1, \dots, N_T\end{cases}$$

where $\mathbf{H}_S^i \in \mathbb{R}^{1\times L}, \boldsymbol{\xi}_S^i \in \mathbb{R}^{1\times m}, \mathbf{t}_S^i \in \mathbb{R}^{1\times m}$ denote the output of hidden layer, the prediction error and the label w.r.t. the *i*-th training instance $\mathbf{x}_S^i$ from the source domain, $\mathbf{H}_T^j \in \mathbb{R}^{1\times L}, \boldsymbol{\xi}_T^j \in \mathbb{R}^{1\times m}, \mathbf{t}_T^j \in \mathbb{R}^{1\times m}$ denote the output of hidden layer, the prediction error and the label vector with respect to the *j*-th guide samples $\mathbf{x}_T^j$ from the target domain, $\boldsymbol{\beta}_S \in \mathbb{R}^{L\times m}$ is the output weights being solved.

Source domain based: DAELM-S

Algorithm for DAELM-S

The Largange multiplier equation is formulated as

$$L(\boldsymbol{\beta}_S, \boldsymbol{\xi}_S^i, \boldsymbol{\xi}_T^j, \alpha_S, \alpha_T) =$$

$$\frac{1}{2}\|\boldsymbol{\beta}_S\|^2 + \frac{C_S}{2}\sum_{i=1}^{N_S}\left\|\boldsymbol{\xi}_S^i\right\|^2 + \frac{C_T}{2}\sum_{j=1}^{N_T}\left\|\boldsymbol{\xi}_T^j\right\|^2 - \boldsymbol{\alpha}_S(\mathbf{H}_S^i\boldsymbol{\beta}_S - \mathbf{t}_S^i + \boldsymbol{\xi}_S^i) - \boldsymbol{\alpha}_T(\mathbf{H}_T^i\boldsymbol{\beta}_T - \mathbf{t}_T^i + \boldsymbol{\xi}_T^i)$$

By setting the partial derivation with respect to $\boldsymbol{\beta}_S, \boldsymbol{\xi}_S^i, \boldsymbol{\xi}_T^j, \boldsymbol{\alpha}_S, \boldsymbol{\alpha}_T$ as zero, we have

$$\begin{cases} \frac{\partial L}{\partial \boldsymbol{\beta}_S} = 0 \rightarrow \boldsymbol{\beta}_S = \mathbf{H}_S^{\mathrm{T}}\boldsymbol{\alpha}_S + \mathbf{H}_T^{\mathrm{T}}\boldsymbol{\alpha}_T \\ \frac{\partial L}{\partial \boldsymbol{\xi}_S} = 0 \rightarrow \boldsymbol{\alpha}_S = C_S\boldsymbol{\xi}_S^{\mathrm{T}} \\ \frac{\partial L}{\partial \boldsymbol{\xi}_T} = 0 \rightarrow \boldsymbol{\alpha}_T = C_T\boldsymbol{\xi}_T^{\mathrm{T}} \\ \frac{\partial L}{\partial \boldsymbol{\alpha}_S} = 0 \rightarrow \mathbf{H}_S\boldsymbol{\beta}_S - \mathbf{t}_S + \boldsymbol{\xi}_S = 0 \\ \frac{\partial L}{\partial \boldsymbol{\alpha}_T} = 0 \rightarrow \mathbf{H}_T\boldsymbol{\beta}_S - \mathbf{t}_T + \boldsymbol{\xi}_T = 0 \end{cases}$$

Source domain based: DAELM-S

Algorithm 1. DAELM-S

Input:

Training samples of the source domain *S*;

Labeled guide samples of the target domain *T*;

The tradeoff parameter C_S and C_T for source and target domain.

Output:

The output weights ;

The predicted output of unlabeled data in target domain.

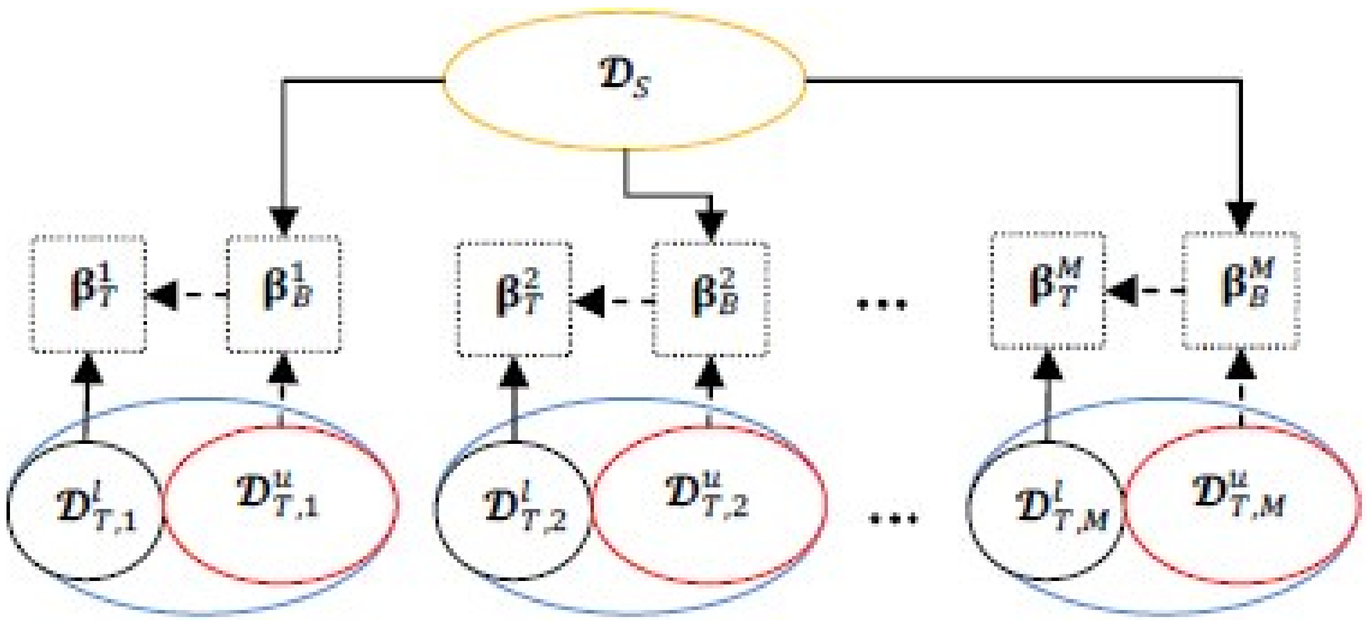
Procedure:

1. Initialize the ELM network of *L* hidden neurons with random input weights **W** and hidden bias **B**.
2. Calculate the output matrix of hidden layer with source and target domains.
3. **If** $N_S<L$, compute the output weights using (12);
   Else, compute the output weights using (13).
4. Calculate the predicted output using (16).

Return The output weights and predicted output .

Target domain based: DAELM-T

- DAELM-T aims to learn a classifier on a very limited number of labeled samples from target domain, by leveraging numerous unlabeled data in target domain, into which a base classifier trained by source data is incorporated.



Target domain based: DAELM-T

- Formulation of DAELM-T

$$\min_{\boldsymbol{\beta}_T} L_{DAELM-T}(\boldsymbol{\beta}_T) = \frac{1}{2}\|\boldsymbol{\beta}_T\|^2 + C_T\frac{1}{2}\|\mathbf{t}_T - \mathbf{H}_T\boldsymbol{\beta}_T\|^2 + C_{Tu}\frac{1}{2}\|\mathbf{H}_{Tu}\boldsymbol{\beta}_B - \mathbf{H}_{Tu}\boldsymbol{\beta}_T\|^2$$

where $\boldsymbol{\beta}_T$ denotes the learned classifier, $C_T, \mathbf{H}_T, \mathbf{t}_T$ are the same as that in DAELM-S, $C_{Tu}, \mathbf{H}_{Tu}$ denote the regularization parameter and the output matrix of the hidden layer with respect to the unlabeled data in target domain.

Regularized ELM is used for training the base classifier $\boldsymbol{\beta}_B$ as

$$\min_{\boldsymbol{\beta}_B} L_{ELM}(\boldsymbol{\beta}_B) = \frac{1}{2}\|\boldsymbol{\beta}_B\|^2 + C_S\frac{1}{2}\|\mathbf{t}_S - \mathbf{H}_S\boldsymbol{\beta}_B\|^2$$

Target domain based: DAELM-T

Algorithm for DAELM-T

The algorithm can be implemented as the similar way of DAELM-S. that uses Largange multiplier method and solve the equations.

$$L(\boldsymbol{\beta}_T, \boldsymbol{\xi}_T^i, \boldsymbol{\xi}_{Tu}^i, \boldsymbol{\alpha}_T, \boldsymbol{\alpha}_{Tu}) = \frac{1}{2}\|\boldsymbol{\beta}_T\|^2 + \frac{C_T}{2}\sum_{i=1}^{N_T}\left\|\boldsymbol{\xi}_T^i\right\|^2 + \frac{C_{Tu}}{2}\sum_{j=1}^{N_{Tu}}\left\|\boldsymbol{\xi}_{Tu}^j\right\|^2 - \boldsymbol{\alpha}_T(\mathbf{H}_T^i\boldsymbol{\beta}_T - \mathbf{t}_T^i + \boldsymbol{\xi}_T^i) - \boldsymbol{\alpha}_{Tu}(\mathbf{H}_{Tu}^i\boldsymbol{\beta}_T - \mathbf{t}_{Tu}^i + \boldsymbol{\xi}_{Tu}^i)$$

where $\mathbf{t}_{Tu} = \mathbf{H}_{Tu}\boldsymbol{\beta}_B$.

By setting the partial derivation with respect to $\boldsymbol{\beta}_T, \boldsymbol{\xi}_T^i, \boldsymbol{\xi}_{Tu}^j, \boldsymbol{\alpha}_T, \boldsymbol{\alpha}_{Tu}$ to be zero, and calculate $\boldsymbol{\beta}_T$

Target domain based: DAELM-T

Algorithm 2. DAELM-T

Input:

Training samples of the source domain S;

Labeled guide samples of the target domain T;

Unlabeled samples of the target domain T;

The tradeoff parameters C_S, C_T and C_{Tu}.

Output:

The output weights ;

The predicted output of unlabeled data in target domain.

Procedure:

1. Initialize the ELM network of L hidden neurons with random input weights $\mathbf{W}_1$ and hidden bias $\mathbf{B}_1$.
2. Calculate the output matrix $\mathbf{H}_S$ of hidden layer with source domain as .
3. **If** N_S<L, compute the output weights of the base classifier using (4);
   Else, compute the output weights of the base classifier using (3).
4. Initialize the ELM network of L hidden neurons with random input weights $\mathbf{W}_2$ and hidden bias $\mathbf{B}_2$.
5. Calculate the hidden layer output matrix and of labeled and unlabeled data in target domains as and.
6. **If** N_T<L, compute the output weights using (25);
   Else, compute the output weights using (20).
7. Calculate the predicted output using (26).

Return The output weights and predicted output.

Remarks on DAELM-S and DAELM-T

- *Remark 1*

The proposed DAELM framework inherits the merits of ELM: 1) random feature mapping; 2) analytically determine the weights. It also draws some new perspective for ELM in multi-domains.

- *Remark 2*

DAELM-S has similar structure in model and algorithm with DAELM-T.

The essential difference lies in that numerous unlabeled data which may be useful for improving generalization performance are exploited in DAELM-T through a pre-learned base classifier.

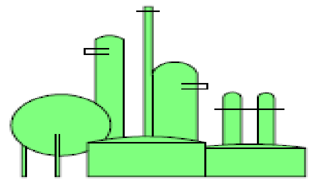
DAELM-S: The main knowledge is from "source domain";

DAELM-T: The main knowledge is from "target domain".

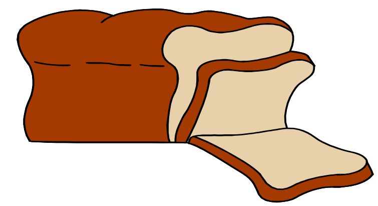
About E-Sensing/E-Olfaction

- Electronic Nose (E-Nose): Artificial Olfaction System

Medical diagnosis

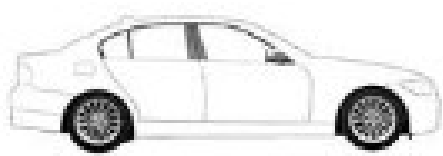


Industrial Monitoring and Process Control
Identity and condition of raw materials, leaks and buildup of toxic compounds; Monitor food processing



Food Safety

MILITARY APPLICATIONS
air quality monitor, detection of explosives and other hazards



Air quality monitor in car

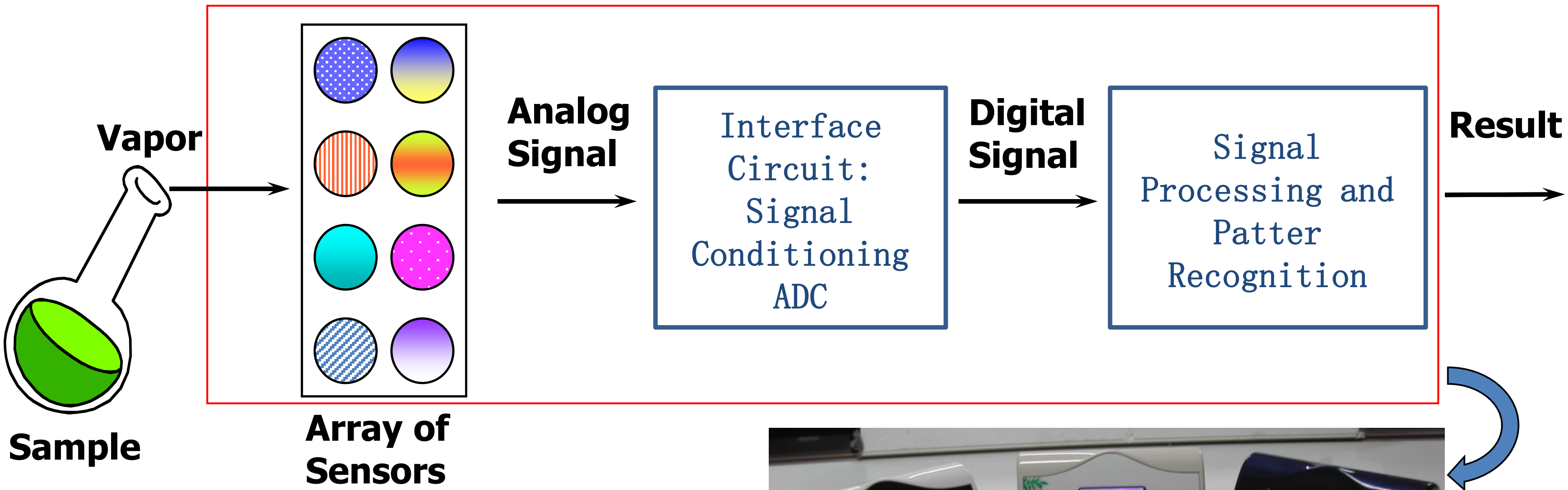
Environmental Monitoring
Air quality in buildings, aircraft. Presence of toxic materials in enclosed spaces (mines, tunnels, etc.)

[3] L. Zhang and F. Tian, "Performance Study of Multilayer Perceptrons in a Low-Cost Electronic Nose," IEEE Transactions on Instrumentation and Measurement, vol. 63, no. 7, Jul. 2014.

[4] L. Zhang, F. Tian, et al. "Classification of multiple indoor air contaminants by an electronic nose and a hybrid support vector machine," Sensors Actu. B, vol. 174, Nov. 2012.

About E-Sensing/E-Olfaction

Electronic Nose (E-Nose): Artificial Olfaction System



[5] L. Zhang, F. Tian, et al. "Gases concentration estimation using heuristics and bio-inspired optimization models for Experimental chemical electronic nose," Sensors. Actu. B., vol. 160, no. 1, Dec. 2011.

Electronic nose instruments developed in our lab

Portable E-NOSE

Experiments

Experimental data

We use the long-term sensor data collected using an E-Nose by Vergara et al [6] .

The sensor drift big dataset was gathered during the period from Jan 2008 to Feb 2011, including 13,910 measurements (observations) from an electronic nose system with 16 gas sensors exposed to 6 kinds of pure gaseous substances.

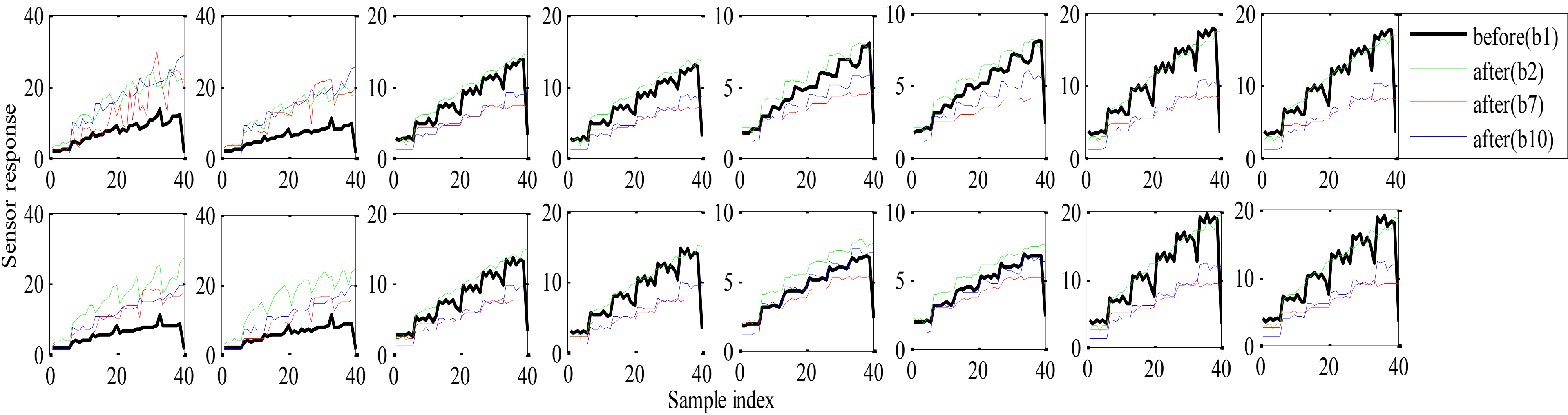
TABLE I
EXPERIMENTAL DATA OF SENSOR DRIFT IN ELECTRONIC NOSE

Batch ID	Month	Acetone	Acetaldehyde	Ethanol	Ethylene	Ammonia	Toluene	Total
Batch 1	1, 2	90	98	83	30	70	74	445
Batch 2	3~10	164	334	100	109	532	5	1244
Batch 3	11, 12, 13	365	490	216	240	275	0	1586
Batch 4	14, 15	64	43	12	30	12	0	161
Batch 5	16	28	40	20	46	63	0	197
Batch 6	17, 18, 19, 20	514	574	110	29	606	467	2300
Batch 7	21	649	662	360	744	630	568	3613
Batch 8	22, 23	30	30	40	33	143	18	294
Batch 9	24, 30	61	55	100	75	78	101	470
Batch 10	36	600	600	600	600	600	600	3600

[6] A. Vergara, S. Vembu, et al., "Chemical gas sensor drift compensation using classifier ensembles," Sensors Actu. B, vol. 166, 2012 . (California Institute of Technology, USA)

Experiments

- Understanding the sensor drift



Experiments

Experimental settings

1) Setting-1: Take batch 1 (source domain) as fixed training set and tested on batch *K*, *K*=2,…,10 (target domains);

2) Setting-2: The training set (source domain) is dynamically changed with batch *K*-1 and tested on batch *K* (target domain), *K*=2,…,10.

Parameter setting

1) In default, the number of hidden neurons *L* is set as 1000, and The RBF function (i.e. *radbas*) with kernel width set as 1 is used as activation function.

2) In DAELM-S, the penalty coefficients C_S and C_T are set as 0.01 and 10; in DAELM-T, C_S and C_T are set as 0.001 and 100.

Experiments

TABLE II
COMPARISONS OF RECOGNITION ACCURACY (%) UNDER THE EXPERIMENTAL SETTING 1

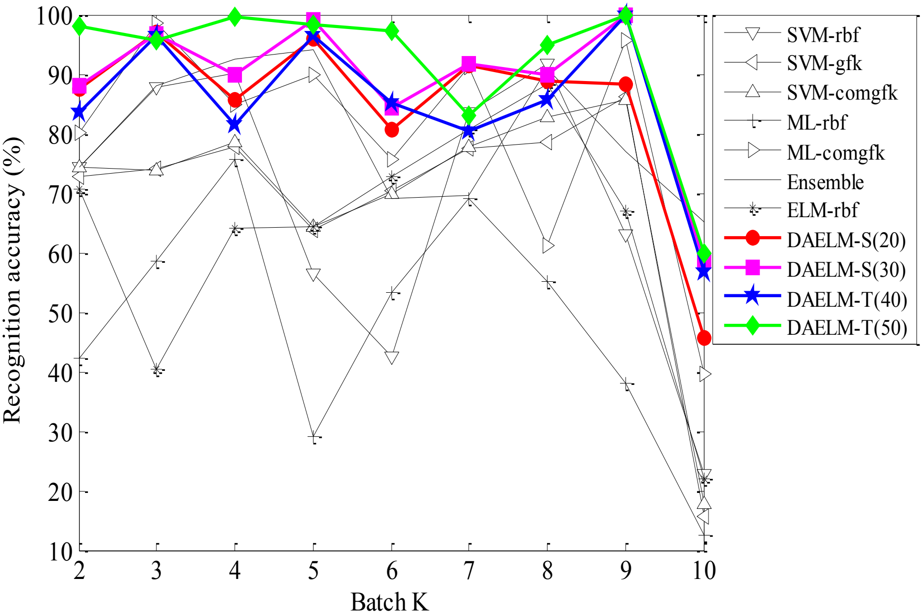
Batch ID	Batch 2	Batch 3	Batch 4	Batch 5	Batch 6	Batch 7	Batch 8	Batch 9	Batch 10	Average
CC-PCA	67.00	48.50	41.00	35.50	55.00	31.00	56.50	46.50	30.50	45.72
SVM-rbf	74.36	61.03	50.93	18.27	28.26	28.81	20.07	34.26	34.47	38.94
SVM-gfk	72.75	70.08	60.75	75.08	73.82	54.53	55.44	69.62	41.78	63.76
SVM-comgfk	74.47	70.15	59.78	75.09	73.99	54.59	55.88	70.23	41.85	64.00
ML-rbf	42.25	73.69	75.53	66.75	77.51	54.43	33.50	23.57	34.92	53.57
ML-comgfk	80.25	74.99	78.79	67.41	77.82	71.68	49.96	50.79	53.79	67.28
ELM-rbf	70.63	66.44	66.83	63.45	69.73	51.23	49.76	49.83	33.50	57.93
Our DAELM-S(20)	87.57	96.53	82.61	81.47	84.97	71.89	78.10	87.02	57.42	**80.84**
Our DAELM-S(30)	87.98	95.74	85.16	95.99	94.14	83.51	86.90	100.0	53.62	**87.00**
Our DAELM-T(40)	83.52	96.34	88.20	99.49	78.43	80.93	87.42	100.0	56.25	**85.62**
Our DAELM-T(50)	97.96	95.34	99.32	99.24	97.03	83.09	95.27	100.0	59.45	**91.86**

TABLE III
COMPARISONS OF RECOGNITION ACCURACY (%) UNDER THE EXPERIMENTAL SETTING 2

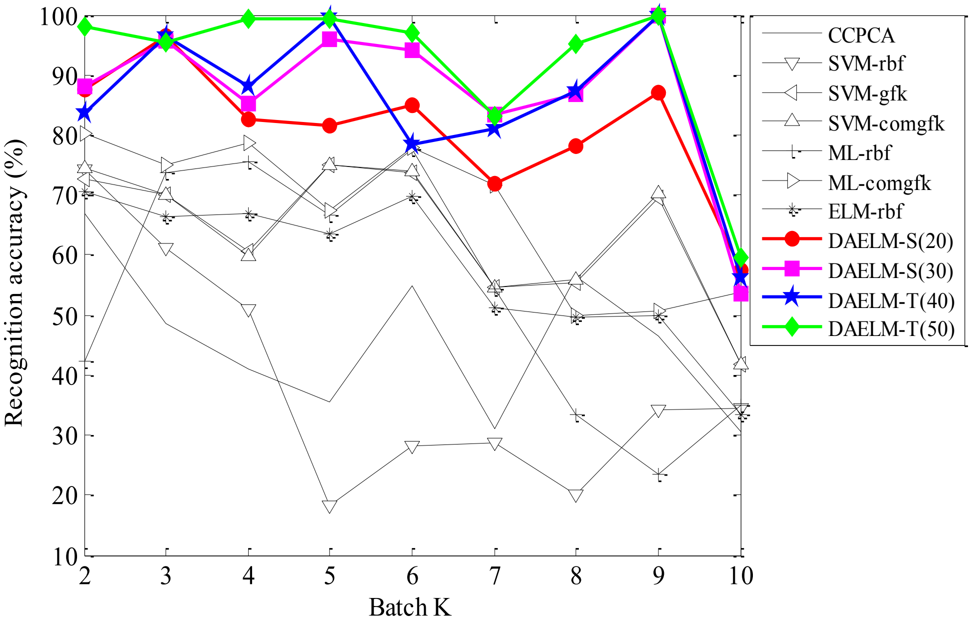
Batch ID	1→2	2→3	3→4	4→5	5→6	6→7	7→8	8→9	9→10	Average
SVM-rbf	74.36	87.83	90.06	56.35	42.52	83.53	91.84	62.98	22.64	68.01
SVM-gfk	72.75	74.02	77.83	63.91	70.31	77.59	78.57	86.23	15.76	68.56
SVM-comgfk	74.47	73.75	78.51	64.26	69.97	77.69	82.69	85.53	17.76	69.40
ML-rbf	42.25	58.51	75.78	29.10	53.22	69.17	55.10	37.94	12.44	48.17
ML-comgfk	80.25	98.55	84.89	89.85	75.53	91.17	61.22	95.53	39.56	79.62
Ensemble	74.40	88.00	92.50	94.00	69.00	69.50	91.00	77.00	65.00	80.04
ELM-rbf	70.63	40.44	64.16	64.37	72.70	80.75	88.20	67.00	22.00	63.36
Our DAELM-S(20)	87.57	96.90	85.59	95.89	80.53	91.56	88.71	88.40	45.61	**84.53**
Our DAELM-S(30)	87.98	96.58	89.75	99.04	84.43	91.75	89.83	100.0	58.44	**88.64**
Our DAELM-T(40)	83.52	96.41	81.36	96.45	85.13	80.49	85.71	100.0	56.81	**85.10**
Our DAELM-T(50)	97.96	95.62	99.63	98.17	97.13	83.10	94.90	100.0	59.88	**91.82**

Experiments

- How about the state of the art on this dataset?



Comparisons of different methods in experimental **Setting 1**

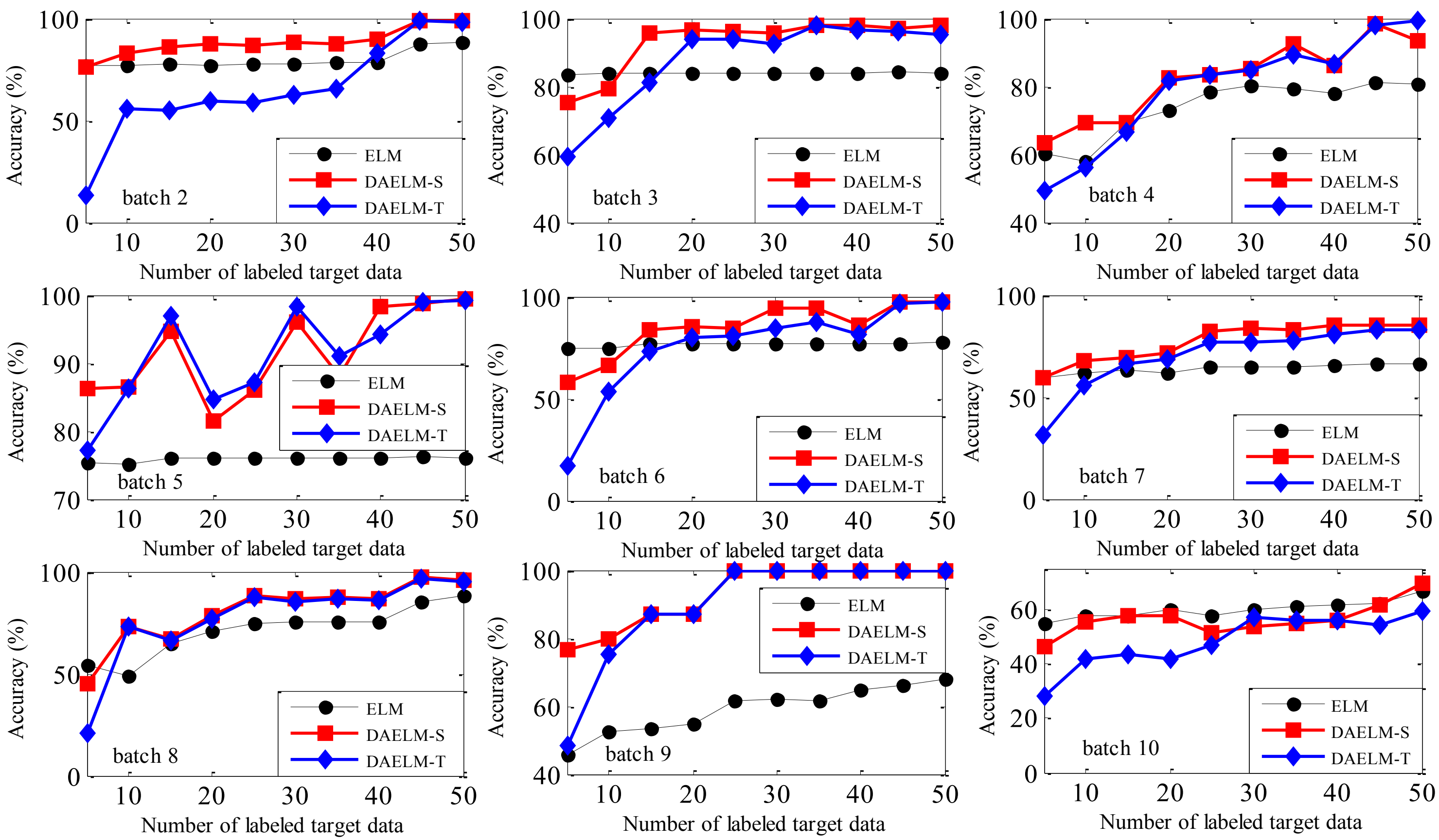


Comparisons of different methods in experimental **Setting 2**

Experiments

What is the difference from ELM?

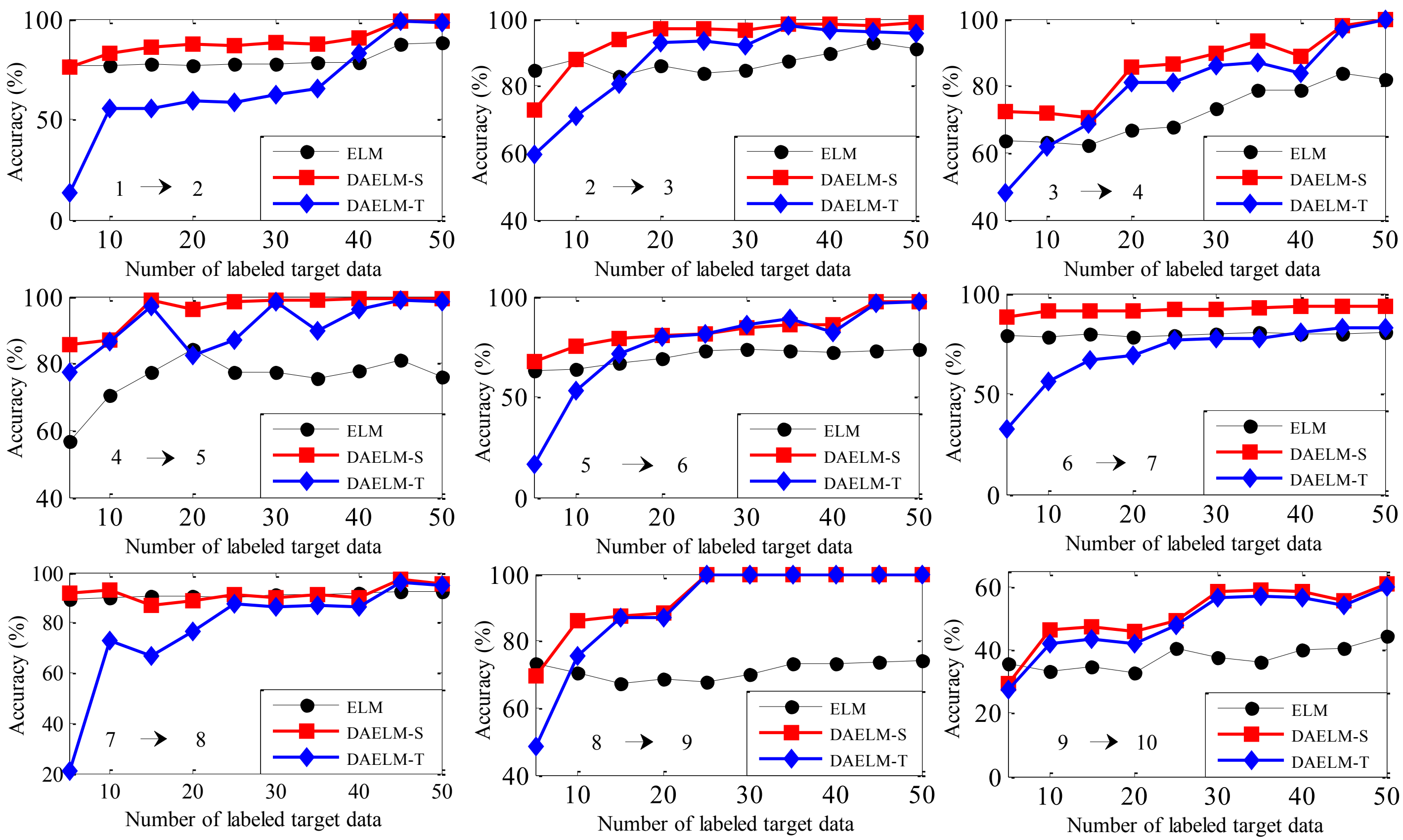
Recognition accuracy under **Setting 1** with respect to different size of guide set (labeled samples from target domain)



Experiments

What is the difference from ELM?

Recognition accuracy under **Setting 2** with respect to different size of guide set (labeled samples from target domain)



Conclusion and future work

- Domain adaptation in transfer learning, as a new perspective, is introduced in ELM theory. The models and algorithms are depicted.
- The proposed methods are verified on a large scale E-sensing data with multiple latent domains.

In near future,

- Applications in Computer Vision Tasks

1) Object recognition in cross domains.
2) Image classification.

- Transfer learning in CV

1) Sparse and low rank reconstruction based subspace transfer.
2) Deep analysis of ELM

Thank you

More information:
http://www.escience.cn/people/lei/index.html