Compact Coding for Hyperplane Classifiers in Heterogeneous Environment

Hao Shao, Bin Tong and Einoshin Suzuki

Kyushu University, Japan

2011/9/5 @ ECML PKDD 2011, Athens

KYUSHU UNIVERSITY 100th 2011
知の新世紀を拓く

Inductive Transfer Learning with multiple source tasks

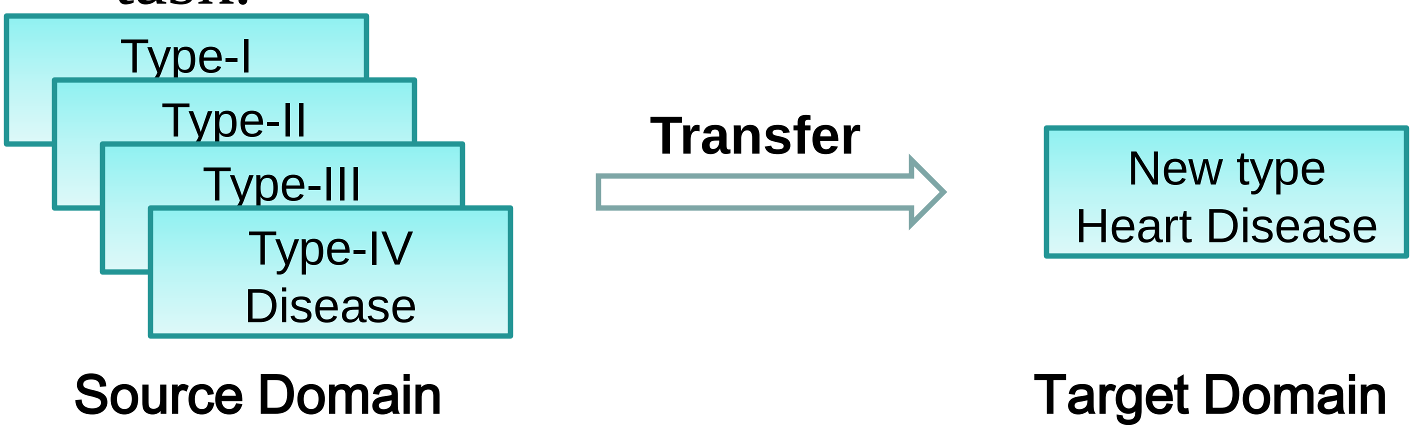
- ***Input***: source data sets S_i (i=1,…,K), target data set T. Each instance $\mathbf{x}$ has the identical nominal attributes set $\{x_1, x_2, \ldots, x_{m-1}\}$, and a class label set {0, 1}.
- ***Output***: A hyperplane classifier of the target task.

only 20 labeled samples

Binary classification problems for heart disease diagnose

Inductive Transfer Learning with multiple source tasks

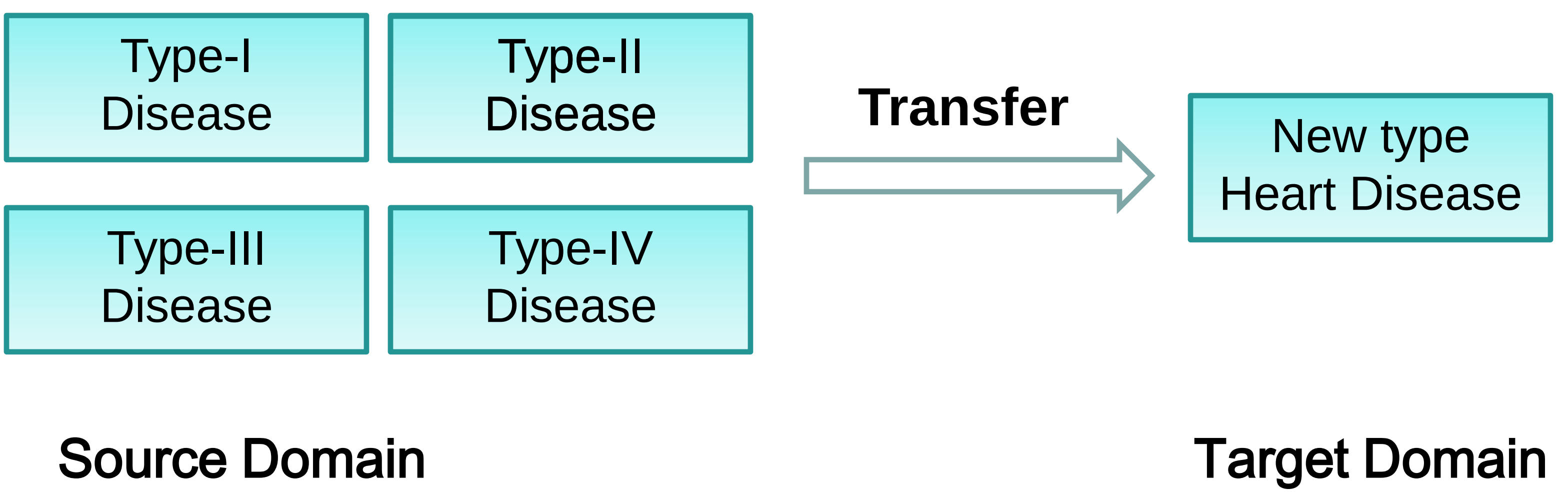
- ***Input***: source data sets S_i (i=1,…,K), target data set T. Each instance $\mathbf{x}$ has the identical nominal attributes set $\{x_1, x_2, \ldots, x_{m-1}\}$, and a class label set {0, 1}.
- ***Output***: A hyperplane classifier of the target task.



Problems of the Negative Transfer (NT)

Two problems

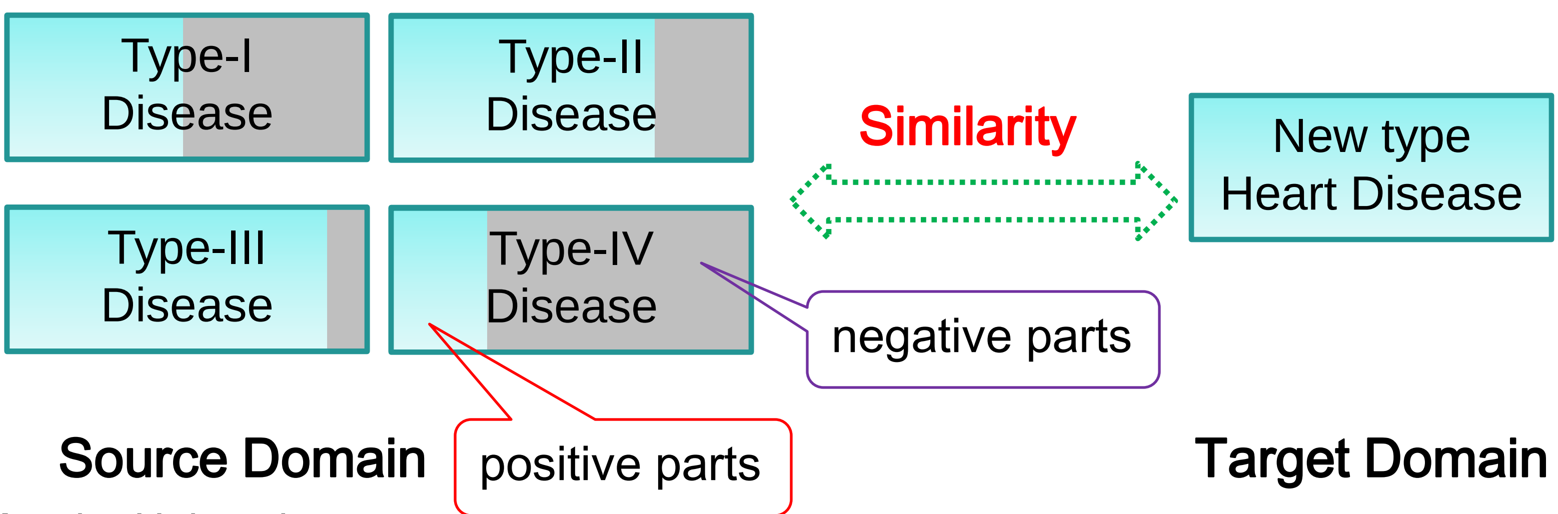
- A source task may be dissimilar with the target task due to the different distributions. Directly transferring knowledge will lead to *Negative Transfer*.
- Not all the data in the similar source tasks are helpful.



Problems of the Negative Transfer (NT)

Two problems

- A source task may be dissimilar with the target task due to the different distributions. Directly transferring knowledge will lead to *Negative Transfer*.
- Not all the data in the similar source tasks are helpful.



Existing Methods and the Objective of our algorithm

- [Cao 10] considered only one source data set.
- Most methods [Argyriou 08, Dai 07] only consider one kind of similarity which is either the similarity between tasks or the similarity between instances.
- Some methods [Dai 07, Shi 08] are heuristic.

Existing Methods and the Objective of our algorithm

- [Cao 10] considered only one source data set.

We consider multiple source tasks

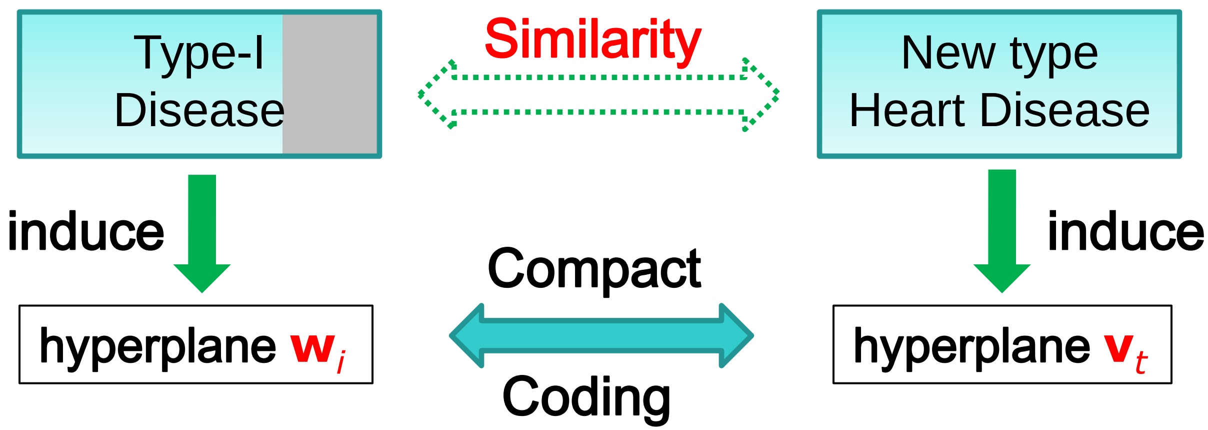
- Most methods [Argyriou 08, Dai 07] only consider one kind of similarity which is either the similarity between tasks or the similarity between instances.

We consider not only the similarity between data sets but also the similarity of different parts within one data set.

- Some methods [Dai 07, Shi 08] are heuristic.

Our method is based on a solid theoretical foundation

Problem Setting and our Motivation

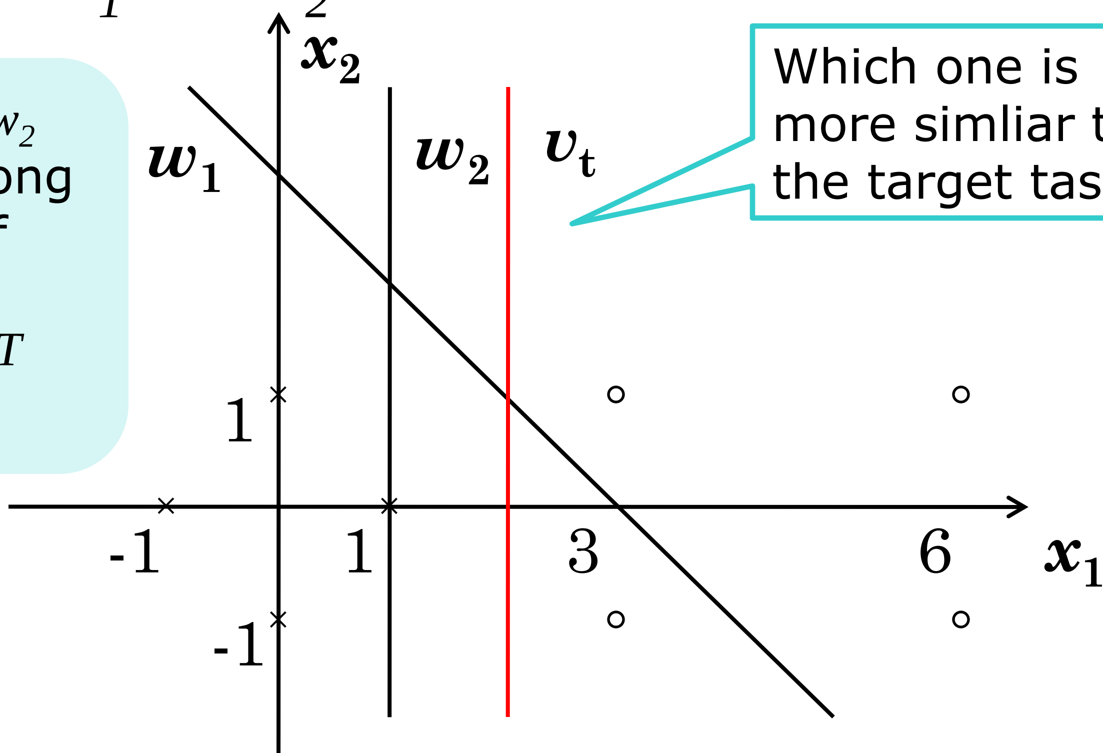


- A hyperplane classifier $\mathbf{w}_i \mathbf{x} = 0$ is induced from each S_i, where $\mathbf{x} = (x_1, x_2, \ldots, x_{m-1}, 1)$, and $\mathbf{w}_i = (w_i^1, w_i^2, \ldots, w_i^m)$. The weight vector of the hyperplane in the target task T is denoted by $\mathbf{v}_t = (v_t^1, v_t^2, \ldots, v_t^m)$.

A Simple Example

- T has 8 labeled examples with hyperplane v_t, w_1 and w_2 are the hyperplanes of two source data sets S_1 and S_2.

Both w_1 and w_2 have one wrong prediction, of the eight examples in T



Which one is more simliar to the target task?

Minimum Description Length Principle (MDLP) [Quinlan 89]

- Best hypothesis: to minimize

  code length of the hypothesis +

  code length of the data using the hypothesis
- Given the data D and the hypothesis h_i (i = 1, 2, 3, …), the best hypothesis h_{best} on D is:

$$h_{best} = \arg\min_{h_i} \left(\underline{-\log P(D \mid h_i) - \log P(h_i)} \right)$$

Balance the simplicity of the hypothesis and the goodness-of-fit to the data

avoid overfitting

Compact Coding for Hyperplane Classifiers (CCHC)

- **Macro Level Evaluation:** Sort S_i in descending order on the degrees of similarity with the target data set T.

- **Micro Level Evaluation:** Divide the data set of the related source tasks into several components and select related parts to help training the classifier in the target domain.

Code Length as the Similarity Measure

A posteriori probability of w_i given the source task S_i : $P(w_i \mid S_i)$

Code Length as the Similarity Measure

$$P(w_i \mid S_i)$$

$$\Downarrow$$

$$P(w_i \mid T)$$

Measure the similarity between w_i and T

Code Length as the Similarity Measure

$$P(w_i \mid S_i)$$

$$\Downarrow$$

$$P(w_i \mid T) \propto P(T \mid w_i)\, P(w_i)$$

Measure the similarity between w_i and T

Code Length as the Similarity Measure

$$P(w_i \mid S_i)$$

$$P(w_i \mid T) \propto P(T \mid w_i)\, P(w_i)$$

$$\Downarrow$$

$$P(w_i \mid T, v_t)$$

Borrow v_t to help to code w_i

Code Length as the Similarity Measure

$$P(w_i \mid S_i)$$

$$P(w_i \mid T) \propto P(T \mid w_i)\, P(w_i)$$

$$\Downarrow$$

$$\begin{aligned} P(w_i \mid T, v_t) &\propto P(T \mid w_i)\, P(v_t \mid w_i)\, P(w_i) \\ &\propto P(T \mid w_i)\, P(w_i \mid v_t)\, P(v_t) \\ &\propto P(T \mid w_i)\, P(w_i \mid v_t) \end{aligned}$$

Borrow v_t to help to code w_i

Code Length as the Similarity Measure

$$P(w_i \mid S_i)$$

$$P(w_i \mid T) \propto P(T \mid w_i)\, P(w_i)$$

$$\begin{aligned} P(w_i \mid T, v_t) &\propto P(T \mid w_i)\, P(v_t \mid w_i)\, P(w_i) \\ &\propto P(T \mid w_i)\, P(w_i \mid v_t)\, P(v_t) \\ &\propto P(T \mid w_i)\, P(w_i \mid v_t) \end{aligned}$$

$$\Downarrow$$

$$\boxed{L_i = -\log P(T \mid w_i) - \log P(v_t \mid w_i)}$$

Preliminaries of coding

- The code length of a binary string of length *a* which consists of *b* binary 1s and (*a*-*b*) binary 0s.

$$\Theta(a,b) \equiv \log(a+1) + \log\binom{a}{b}$$

- Coding a real number *x* under the assumption that $x=\mu$ is most likely, where μ is also a real number, and f is a continuous probability with precision ε.

$$\Lambda(x,\mu) = -\log P(x) = -\log\left(\int_{x-\frac{\varepsilon}{2}}^{x+\frac{\varepsilon}{2}} f(x)dx\right)$$

18

Coding method of CCHC

- The first part of the code length is:

$$-\log P(\mathbf{w}_i \mid \mathbf{v}_t) = \sum_{j=1}^{m} \Lambda(w_i^j, v_t^j)$$

- The second part of the code length is:

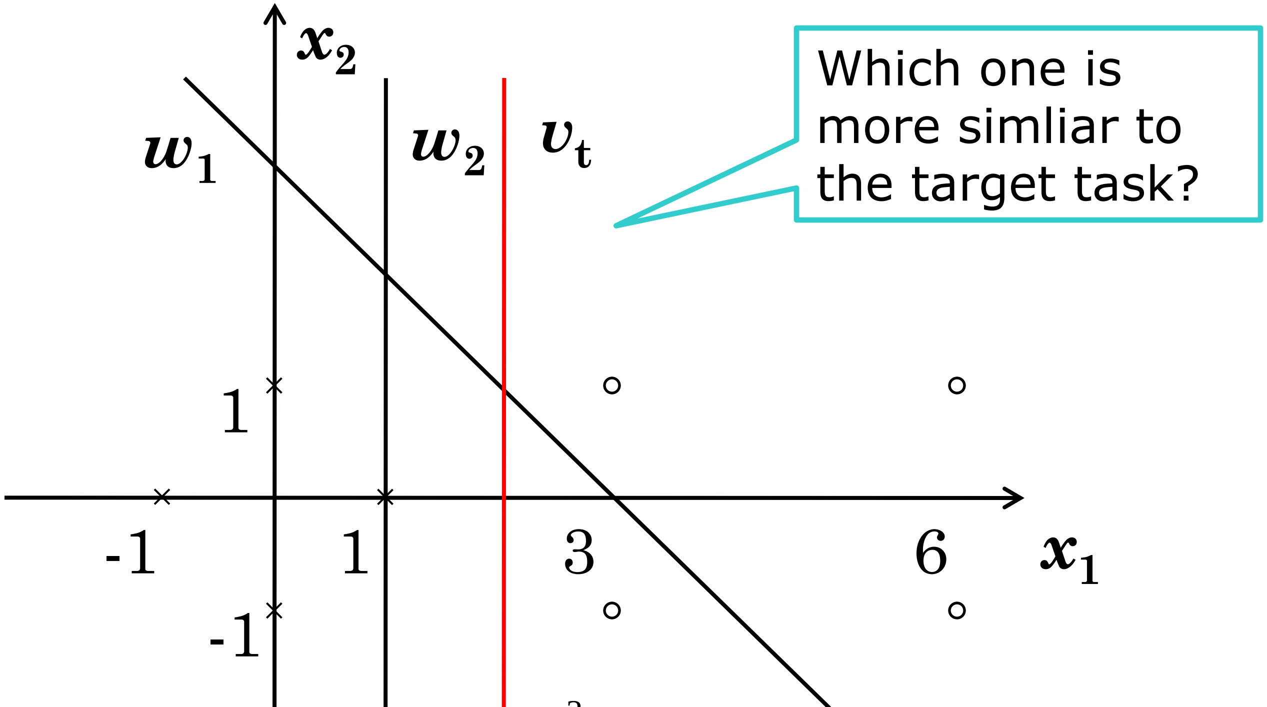
$$-\log P(T \mid \mathbf{w}_i) = \Theta(|T|, \omega(\mathbf{w}_i, T))$$

where $\omega(\mathbf{w}_i, T)$ denotes the number of misclassified examples on T.

The code length as the similarity measure:

$$L_i = \sum_{j=1}^{m} \Lambda(w_i^j, v_t^j) + \Theta(|T|, \omega(\mathbf{w}_i, T))$$

Calculation of the code length of the toy example



$$L_1 = \Theta(|T|, \omega(\mathbf{w}_1, T)) + \sum_{j=1}^{3} \Lambda(w_1^j, v_t^j) = 587.31 bits$$

$$\boxed{L_2 = \Theta(|T|, \omega(\mathbf{w}_2, T)) + \sum_{j=1}^{3} \Lambda(w_2^j, v_t^j) = 297.22 bits}$$

```
Algorithm CCHC
  for i = 1 to K
    calculate L_i for each S_i by (8), obtain L_min
    sort S_i based on the ascending order of L_i
  TR = T
  for j = 1 to K
    perform clustering on S_j, obtain S_j^t (t = 1,...,n_s)
    calculate l_t for each S_j^t by (8)
    sort S_j^t based on the ascending order of l_t
    for t = 1 to n_s
      TR = TR ∪ S_j^t with the shortest l_t
      perform classification by SVM on TR and obtain w'
      calculate L' = − log P(w'|v_t) − log P(T|w')
      if L' < L_min
        L_min = L'
        v_t = w'
        S_j = S_j − S_j^t
      else break
  w_t = v_t
  output w_t
```

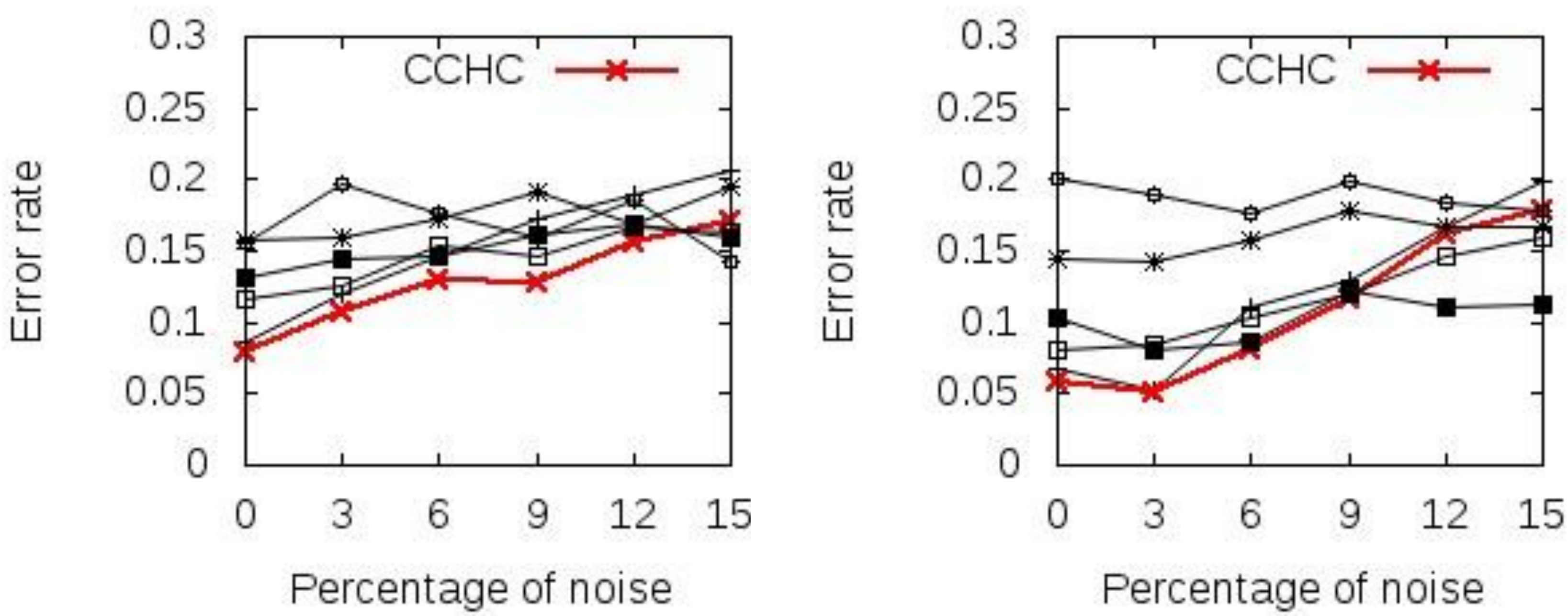
Macro Level

Micro Level

Experimental setting

- Data sets
  - **UCI data sets:** Three data sets are used in the experiments in UCI repository. A pre-processing method [Y. Shi 09] is adopted on these data sets to split each data to the source and the target data sets.
  - **Text data sets:** 20NewsGroup Data sets in three categories, with pre-processing method given by [W. Dai 07] to form different tasks with subcategories.
- State-of-the-art methods for comparison
  - SVM, TrAdaBoost, *k*-NN, COITL [Y. Shi 09] and AT [X. Shi 08].

Results on *mushroom* data sets

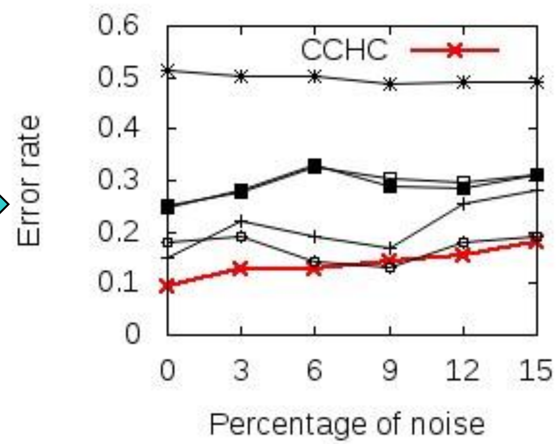


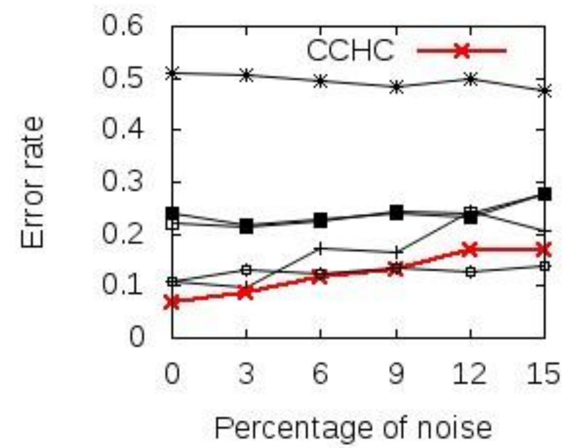
$|T| = 50$ $|T| = 100$

Our method is able to achieve lower error rate with only few labeled information available.

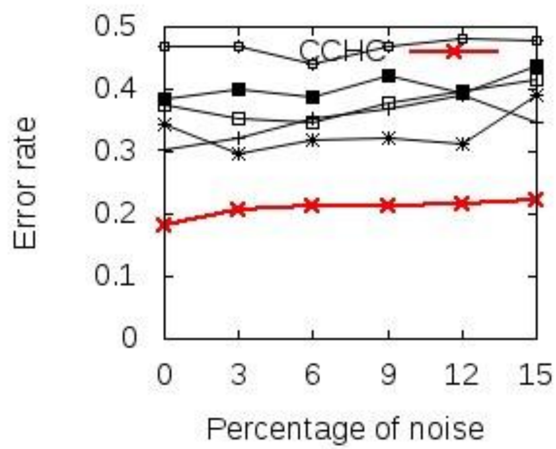
Results of *kr vs kp* and *splice*

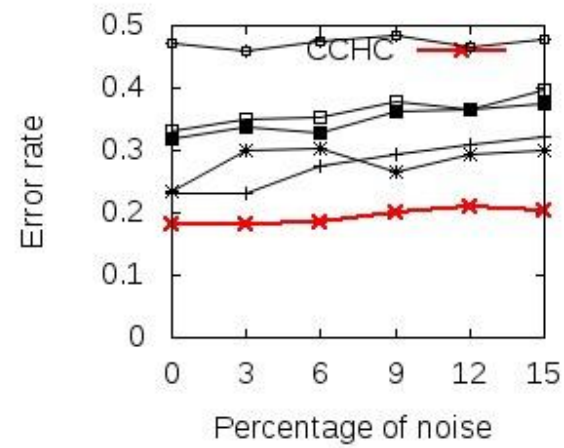
kr vs kp





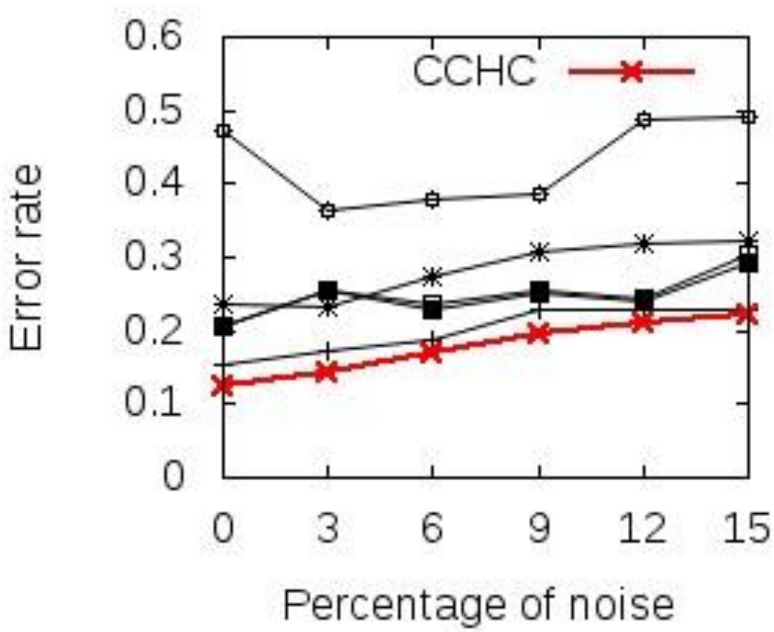
splice



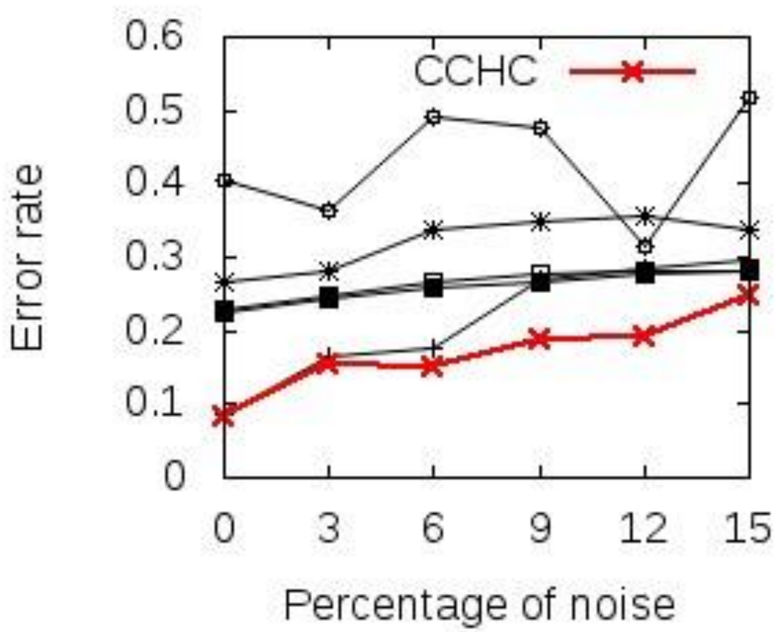


$|T| = 50$ $|T| = 100$

Results for *rec vs talk* as the target data set



$|T| = 50$



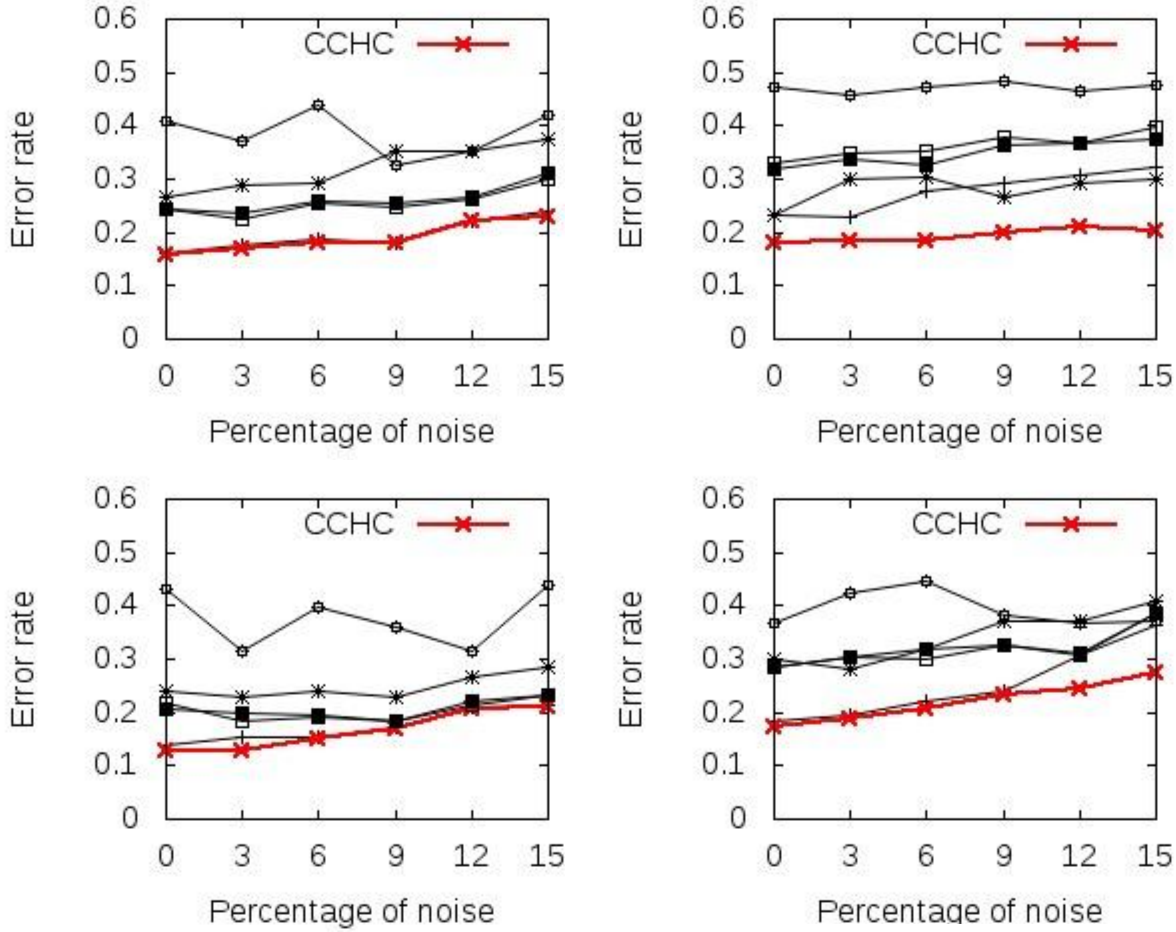
$|T| = 100$

Our method is the best one among all methods.

Results for *rec vs sci*, and *talk vs sci* as the target data set

rec vs sci

talk vs sci



$|T| = 50$ $|T| = 100$

Transferred components in text data sets in Micro Level

Source Data Sets

S_1 : *rec vs talk* S_2 : *rec vs sci* S_3 : *sci vs talk*

			Percentage of noise on T 0%	3%	6%	9%	12%	15%
rec vs *talk* as T	$\|T\| = 50$	S_1	0	0	0	1	0	$\boxed{0}$
		S_2	1	1	1	1	1	0
		S_3	1	1	1	1	1	$\boxed{1}$
	$\|T\| = 100$	S_1	1	0	0	0	0	0
		S_2	1	1	1	1	1	1
		S_3	1	1	1	1	1	1

no parts transferred

1 part transferred

The Micro Level Evaluation is effective which can adaptively select related parts for transferring.

Summary of this work

- Motivation: Design a coding method for hyperplane classifiers in transfer learning. Adaptively select related parts in the source tasks in classifying the target task.
- Methodology: We propose a compact coding method inspired by MDLP, to measure the similarity between data by the code length.
- Performance: Experiments conducted on both UCI and text data sets show the effectiveness of our CCHC.

THANK YOU

Hao Shao, Bin Tong and Einoshin Suzuki

Kyushu University, Japan