LAMDA
Learning And Mining from DatA
http://lamda.nju.edu.cn

南京大學

Semi-Supervised Learning

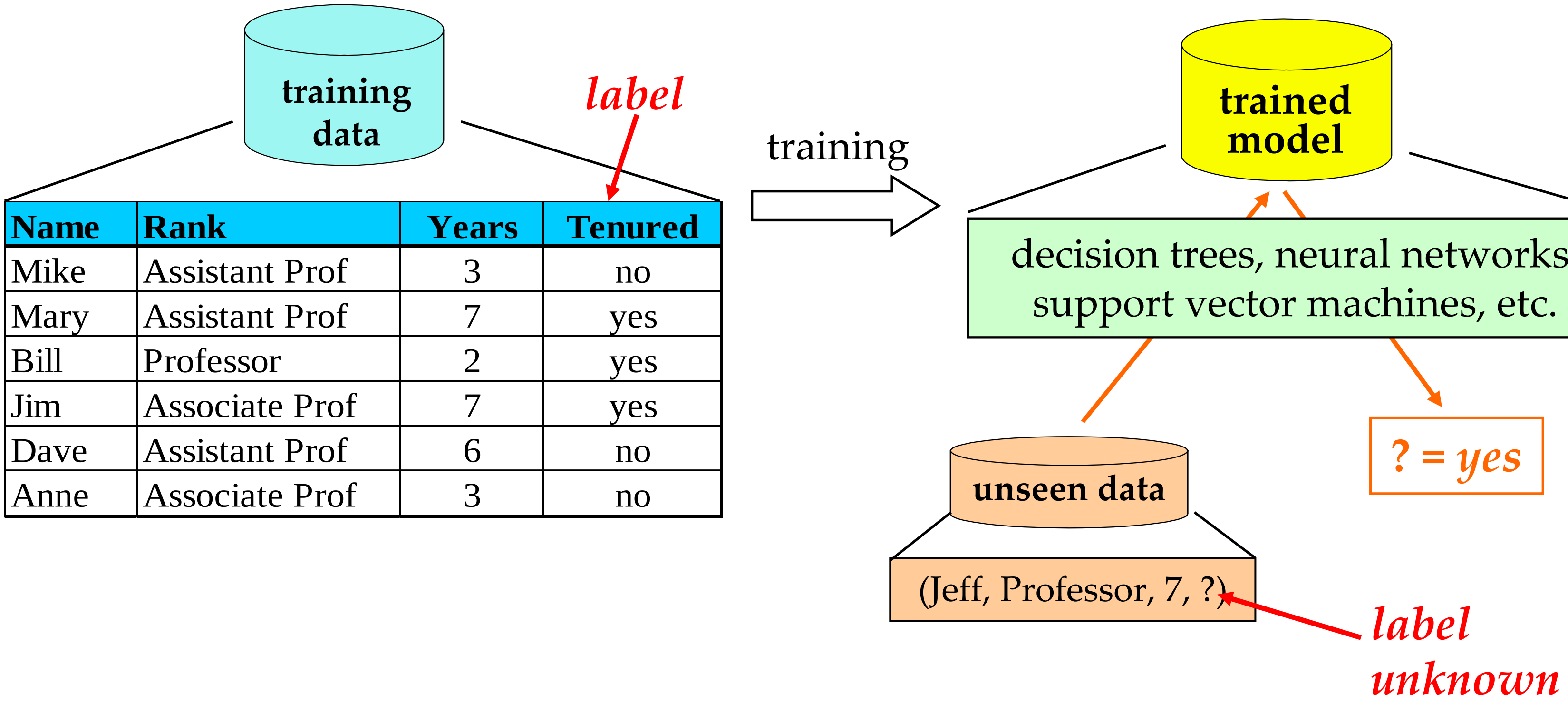
Zhi-Hua Zhou

http://cs.nju.edu.cn/people/zhouzh/
zhouzh@nju.edu.cn

LAMDA Group,
National Laboratory for Novel Software Technology,
Nanjing University, China

Supervised learning

Supervised learning is a typical machine learning setting, where *labeled* examples are used as training examples



Name	Rank	Years	Tenured
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

How to exploit unlabeled data?

In many practical applications, unlabeled training examples are readily available but labeled ones are fairly expansive to obtain because labeling the unlabeled examples requires human effort

Two popular schemes for exploiting unlabeled data to help supervised learning:

- Semi-supervised learning: the learner tries to exploit the unlabeled examples by itself
- Active learning: the learner actively selects some unlabeled examples to query from an *oracle* (assume the learner has some control over the input space)

Why unlabeled training data can be helpful?

An analysis [D.J. Miller & H.S. Uyar, NIPS96]

Suppose the data is well-modeled by a mixture density:

$$f\left(x|\theta\right)=\sum_{l=1}^{L}\alpha_l f\left(x|\theta_l\right) \quad \text{where} \sum\nolimits_{l=1}^{L}\alpha_l=1 \text{ and } \theta=\{\theta_l\}$$

The class labels are viewed as random quantities and are assumed chosen conditioned on the selected mixture component $m_i \in \{1,2,\ldots,L\}$ and possibly on the feature value, i.e. according to the probabilities $\mathrm{P}[c_i | x_i, m_i]$

Thus, the optimal classification rule for this model is the MAP rule:

$$S\left(x\right)=\arg\max_{k}\sum\nolimits_{j}\mathrm{P}\left[c_i=k|m_i=j,x_i\right]\mathrm{P}\left[m_i=j|x_i\right]$$

$$\text{where } \mathrm{P}\left[m_i=j|x_i\right]=\frac{\alpha_j f\left(x_i|\theta_j\right)}{\sum_{l=1}^{L}\alpha_l f\left(x_i|\theta_l\right)}$$

unlabeled examples can be used to help estimate this term

Recognition of the usefulness of unlabeled training examples

The usefulness of unlabeled training examples was recognized by R.P. Lippmann [IEEE ComMag89]:

> "Classifiers that use combined unsupervised/supervised training typically first use unsupervised training with unlabeled data to form internal clusters. Labels are then assigned to clusters and cluster centroid locations, and sizes are often altered using a small amount of supervised training data", "Although combined unsupervised/supervised training mimics some aspects of biological learning, it is of interest primarily because it can reduce the amount of labeled training data required"

"However, despite this realization, there has been surprisingly little work done on this problem. One likely reason is that it does not appear possible to incorporate unlabeled data directly within conventional supervised learning methods such as back propagation."

[D.J. Miller & H.S. Uyar, NIPS96]

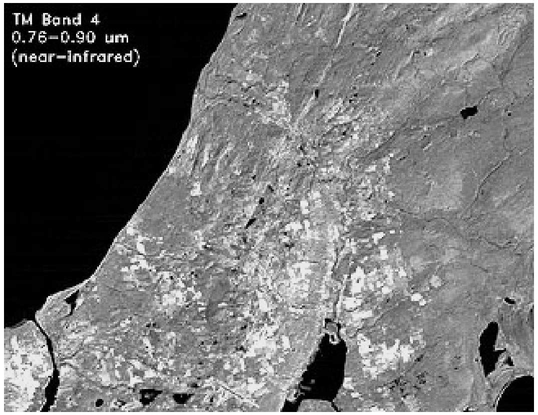
The earliest work

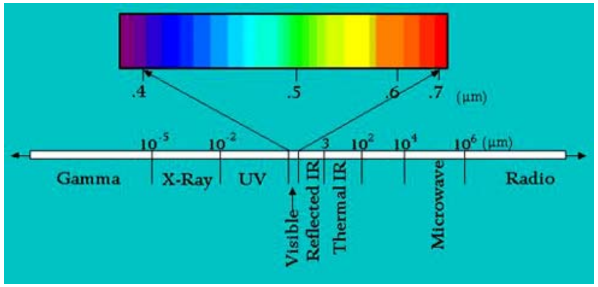
[B.M. Shahshahani & D.A. Landgrebe, TGRS94]

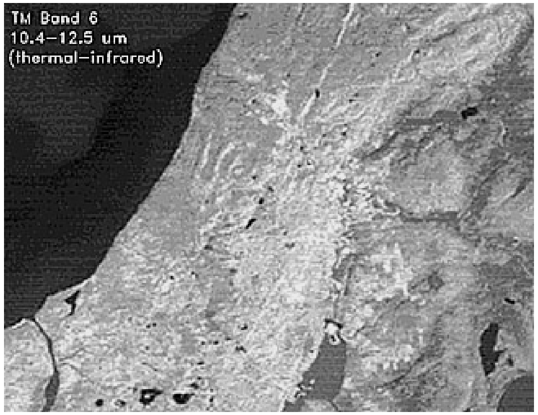
In remote sensing, ground truth information must be gathered by visual inspection of *the scene* near the same time that the data is being taken

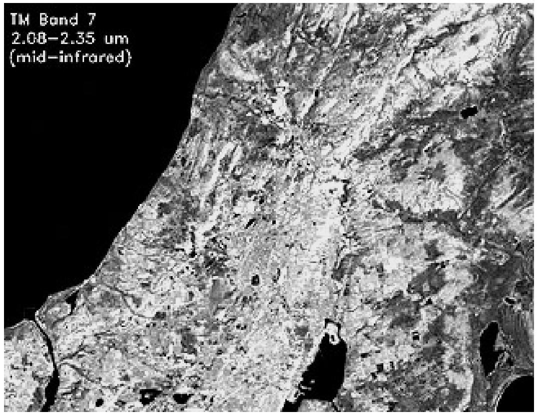
Landsat TM
RGB=321

TM Band 1
0.45–0.52 um
(blue)









The earliest work (con’t)

[B.M. Shahshahani & D.A. Landgrebe, TGRS94]

B.M Shahshahani & D.A. Landgrebe employed Mixture of Gaussians to model the pdf of each class, and then employed EM to estimate the parameters using both labeled and unlabeled training examples (their experimental results were obtained by using single Gaussian to model the pdf of each class)

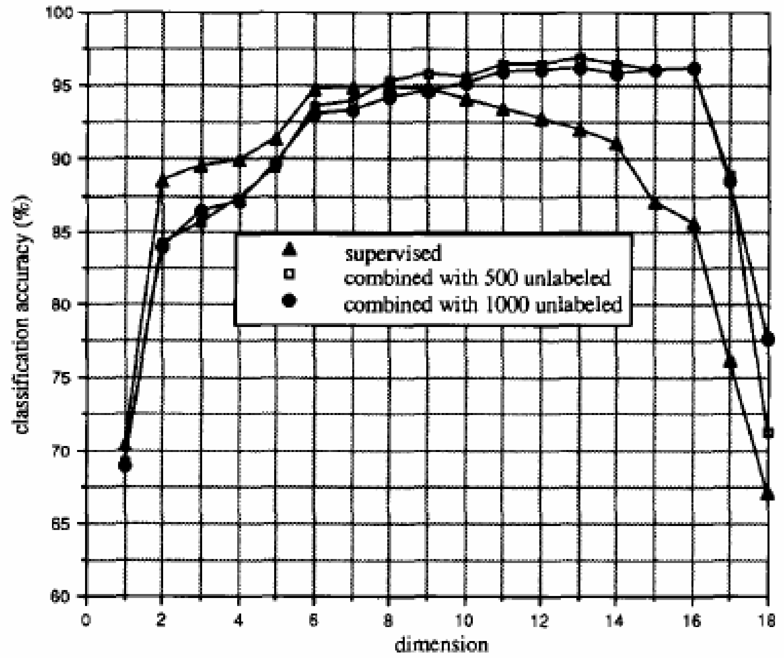


Fig. 4. Effect of additional unlabeled samples in the classification performance for experiment 1 (AVIRIS data) with 20 training samples/class.

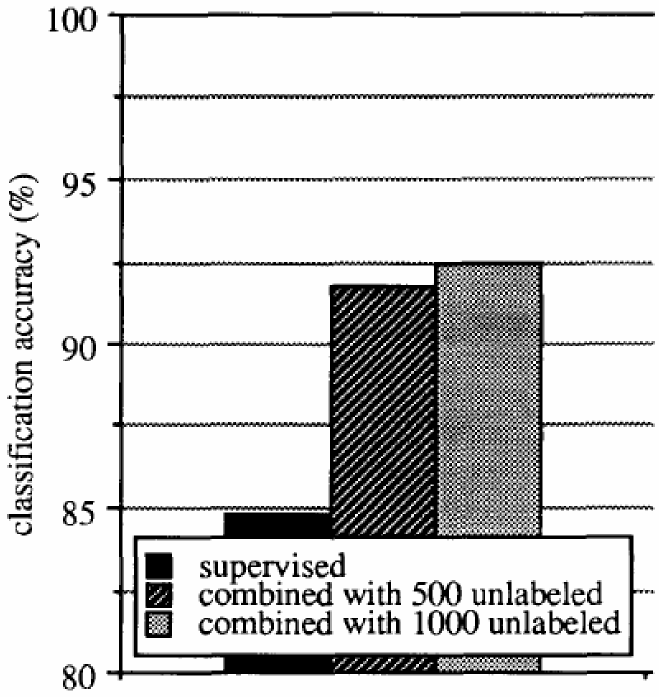


Fig. 6. Classification results based on adjacent training samples.

Figures reprinted from [B.M. Shahshahani & D.A. Landgrebe, TGRS94]

Representative approaches

- Generative model + EM

  Different kinds of generative models have been used, e.g.

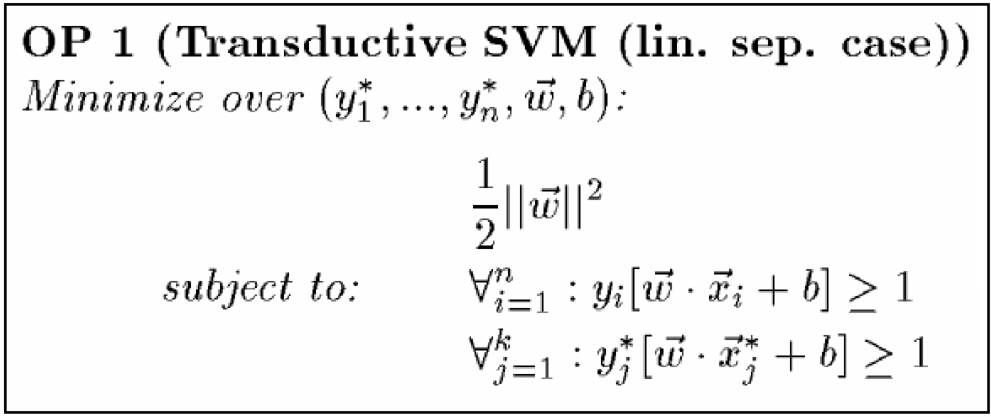
  - mixture of Gaussians [B.M Shahshahani & D.A. Landgrebe, TGRS94]
  - mixture of experts [D.J. Miller & H.S. Uyar, NIPS96]
  - naive Bayes [K. Nigam et al., MLJ00]

- Transductive inference [T. Joachims, ICML99]
- Graph-cut-based [A. Blum & S. Chawla, ICML01]
- Co-training [A. Blum & T. Mitchell, COLT98]
- … …

Learning with transductive SVM

[T. Joachims, ICML99]

For linearly separable problems [V. Vapnik, SLT Book 98]:



OP 1 (Transductive SVM (lin. sep. case))
Minimize over $(y_1^*, \ldots, y_n^*, \vec{w}, b)$:

$$\frac{1}{2}||\vec{w}||^2$$

subject to: $\forall_{i=1}^{n} : y_i[\vec{w} \cdot \vec{x}_i + b] \geq 1$
$\forall_{j=1}^{k} : y_j^*[\vec{w} \cdot \vec{x}_j^* + b] \geq 1$

training examples:

$$(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_n, y_n)$$

test examples $\vec{x}_1^*, \vec{x}_2^*, \ldots, \vec{x}_k^*$

Solving this problem means finding a labeling y_1*, …, y_k* of the test data and a hyperplane <$\vec{w}$, b>, so that this hyperplane separates both training and test data with maximum margin

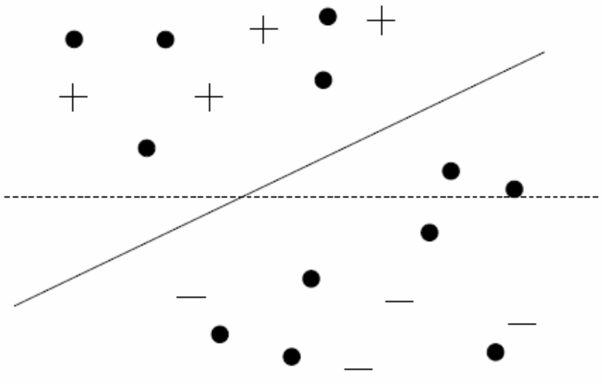


Figure 2: The maximum margin hyperplanes. Positive/negative examples are marked as +/−, test examples as dots. The dashed line is the solution of the inductive SVM. The solid line shows the transductive classification.

Figures reprinted from [T. Joachims, ICML99]

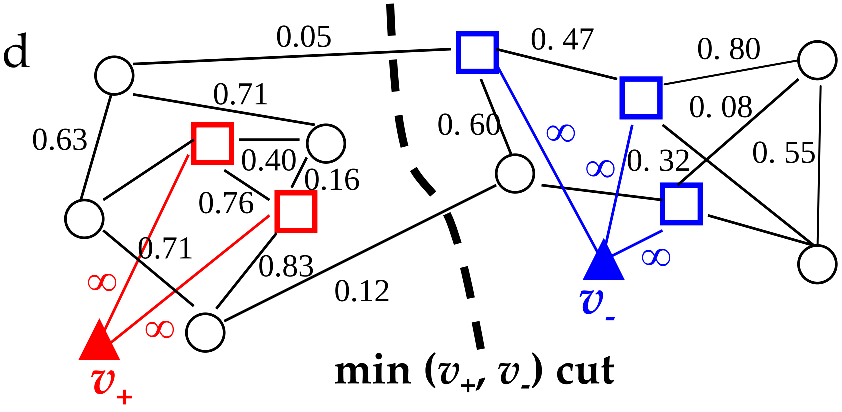
Learning with graph mincuts

[A. Blum & S. Chawla, ICML01]

- Construct a weighted graph on the labeled and unlabeled training examples

  the edge weights correspond to some relationship (such as similarity/distance) between the examples

- To find a minimum (v_+, v_-) cut for the graph
- Assign a positive label to all unlabeled examples in the subgraph of v_+ while a negative label to these in the subgraph of v_-



○ unlabeled example

□ □ labeled examples

▲ ▲ classification vertices

This approach is not as general as some other approaches (such as EM)

the kinds of functions that can be optimized are limited to depend only on pairwise relationships among examples

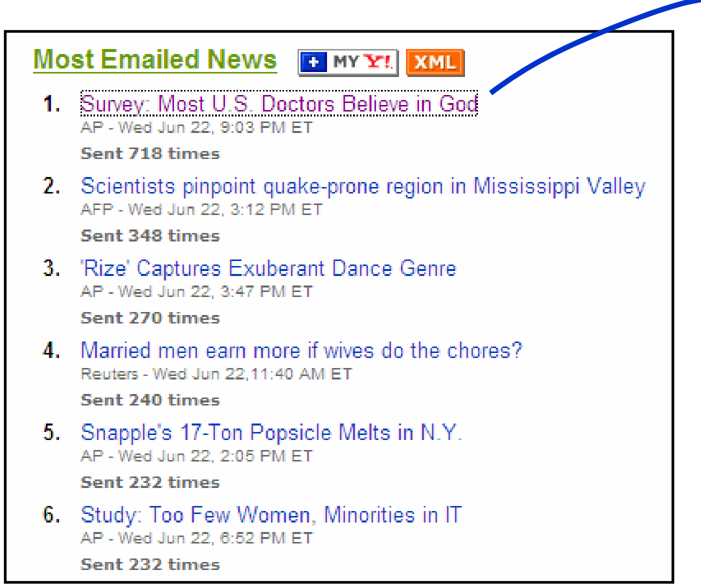
But for the functions it can handle, graph mincuts give a *polynomial-time* algorithm to find the true *global* optimum

Co-training

[A. Blum & T. Mitchell, COLT98]

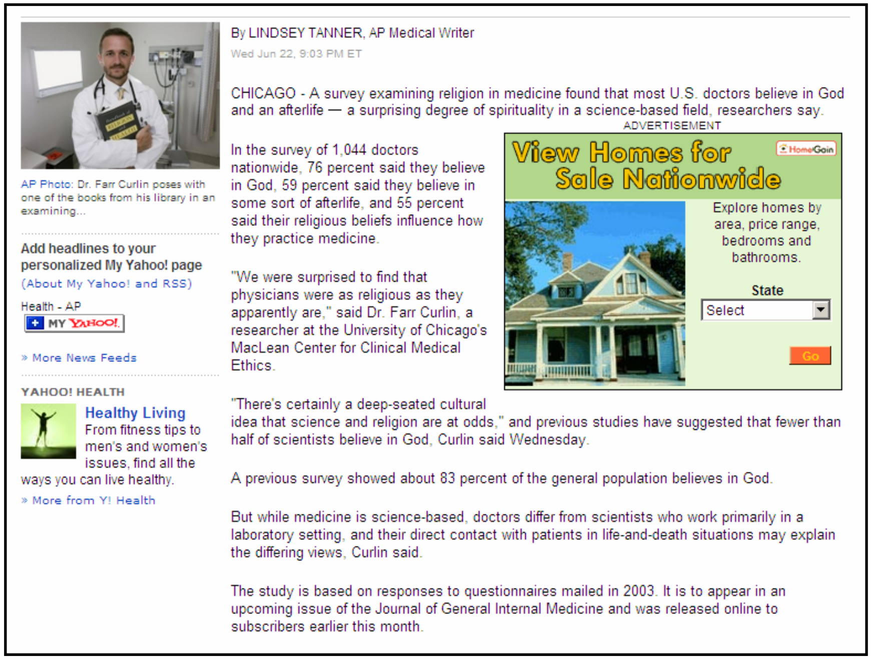
In some applications, there are two sufficient and redundant views, i.e. two attribute sets each of which is sufficient for learning and conditionally independent to the other given the class label

e.g. two views for web page categorization: 1) the text appearing on the page itself, and 2) the anchor text attached to hyperlinks pointing to this page, from other pages



Most Emailed News MY Y! XML

1. Survey: Most U.S. Doctors Believe in God
AP - Wed Jun 22, 9:03 PM ET
Sent 718 times
2. Scientists pinpoint quake-prone region in Mississippi Valley
AFP - Wed Jun 22, 3:12 PM ET
Sent 348 times
3. 'Rize' Captures Exuberant Dance Genre
AP - Wed Jun 22, 3:47 PM ET
Sent 270 times
4. Married men earn more if wives do the chores?
Reuters - Wed Jun 22, 11:40 AM ET
Sent 240 times
5. Snapple's 17-Ton Popsicle Melts in N.Y.
AP - Wed Jun 22, 2:05 PM ET
Sent 232 times
6. Study: Too Few Women, Minorities in IT
AP - Wed Jun 22, 6:52 PM ET
Sent 232 times



AP Photo: Dr. Farr Curlin poses with one of the books from his library in an examining...

Add headlines to your personalized My Yahoo! page
(About My Yahoo! and RSS)

Health - AP
MY YAHOO!

» More News Feeds

YAHOO! HEALTH

Healthy Living
From fitness tips to men's and women's issues, find all the ways you can live healthy.

» More from Y! Health

By LINDSEY TANNER, AP Medical Writer
Wed Jun 22, 9:03 PM ET

CHICAGO - A survey examining religion in medicine found that most U.S. doctors believe in God and an afterlife — a surprising degree of spirituality in a science-based field, researchers say.

In the survey of 1,044 doctors nationwide, 76 percent said they believe in God, 59 percent said they believe in some sort of afterlife, and 55 percent said their religious beliefs influence how they practice medicine.

"We were surprised to find that physicians were as religious as they apparently are," said Dr. Farr Curlin, a researcher at the University of Chicago's MacLean Center for Clinical Medical Ethics.

"There's certainly a deep-seated cultural idea that science and religion are at odds," and previous studies have suggested that fewer than half of scientists believe in God, Curlin said Wednesday.

A previous survey showed about 83 percent of the general population believes in God.

But while medicine is science-based, doctors differ from scientists who work primarily in a laboratory setting, and their direct contact with patients in life-and-death situations may explain the differing views, Curlin said.

The study is based on responses to questionnaires mailed in 2003. It is to appear in an upcoming issue of the Journal of General Internal Medicine and was released online to subscribers earlier this month.

Co-training (con't)

[A. Blum & T. Mitchell, COLT98]

Instance space $X = X_1 \times X_2$, therefore each example is given as a pair (x_1, x_2)

Assume that if f denotes the combined target concept over the entire example, then for any example $x = (x_1, x_2)$ observed with label l, $f(x) = f_1(x_1) = f_2(x_2) = l$

Compatibility:

For a given distribution D over X, where C_1 and C_2 be concept classes defined over X_1 and X_2 respectively, a target function $f = (f_1, f_2) \in C_1 \times C_2$ is said being "compatible" with D if it satisfies the condition that D assigns probability zero to the set of examples (x_1, x_2) such that $f_1(x_1) \neq f_2(x_2)$

Even if C_1 and C_2 are large concept classes with high complexity in, say, the VC-dimension measure, for a given distribution D the set of compatible target concepts might be much simpler and smaller

Thus, one might hope to be able to use unlabeled examples to gain a better sense of which target concepts are compatible, yielding information that could reduce the number of labeled examples needed by a learning algorithm

Co-training (con't)

[A. Blum & T. Mitchell, COLT98]

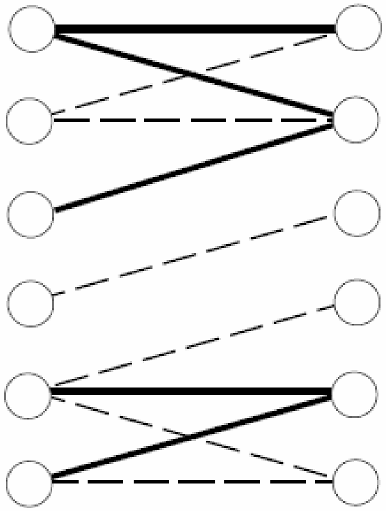


Figure 1: Graphs $G_{\mathcal{D}}$ and G_S. Edges represent examples with non-zero probability under $\mathcal{D}$. Solid edges represent examples observed in some finite sample S. Notice that given our assumptions, even without seeing any labels the learning algorithm can deduce that any two examples belonging to the same connected component in G_S must have the same classification.

One way to look at the co-training problem is to view the distribution *D* as a weighted bipartite graph $G_D(X_1, X_2)$.

The left-hand side of G_D has one node for each point in X_1 and the right-hand side has one node for each point in X_2. There is an edge (x_1, x_2) if and only if the example (x_1, x_2) has non-zero probability under *D*

In this representation, the "compatible" concepts in *C* are exactly those corresponding to a partition of this graph with no cross-edges

Figure reprinted from [A. Blum & T. Mitchell, COLT98]

Co-training (con’t)

[A. Blum & T. Mitchell, COLT98]

Given a *conditional independence* assumption on the distribution *D*, if the target class is learnable from random classification noise in the standard PAC model, then any initial weak predictor can be boosted to arbitrarily high accuracy using *unlabeled examples only* by co-training

A theorem proved by A.Blum & T. Mitchell:

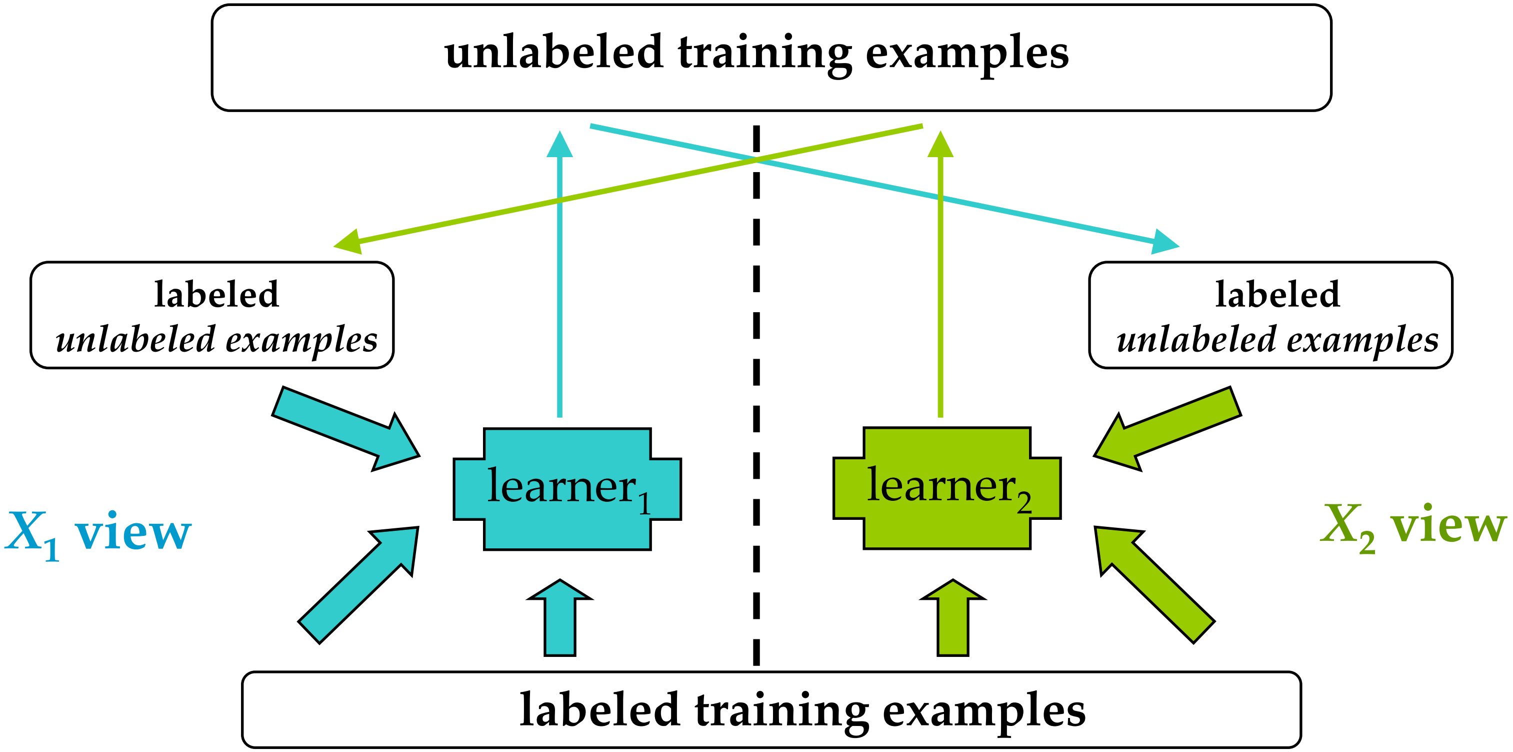
> If C_2 is learnable in the PAC model with classification noise, and if the conditional independence assumption is satisfied, then (C_1, C_2) is learnable in the co-training model from unlabeled data only, given an initial weakly-useful predictor $h(x_1)$

Note:

- *“Co-training” is in fact a paradigm, not a concrete algorithm*
- *It can have different realizations. The one presented by A.Blum & T. Mitchell is called as “the standard co-training algorithm”*

Co-training (con't)

[A. Blum & T. Mitchell, COLT98]



Applications of co-training

Although the requirement of sufficient and redundant views is quite strict, the co-training paradigm has already been used in many domains, e.g.

- Statistical parsing [A. Sarkar, NAACL01; M. Steedman et al., EACL03; R. Hwa et al., ICML03w]
- Noun phrase identification [D. Pierce & C. Cardie, EMNLP01]
- … …

Co-training without two sufficient and redundant views [S. Goldman & Y. Zhou, ICML00]

S. Goldman & Y. Zhou used two different supervised learning algorithm whose hypothesis partitions the example space into a set of equivalent classes

e.g. for a decision tree each leaf defines an equivalent class

in this paper, the ID3 decision tree and HOODG decision tree were used to implement the two classifiers

Two key issues:

- **Combining**: since the classifiers are different, their outputs can hardly be compared directly. Then, how to combine the two classifiers?
- **Choosing which example to label**: when should classifier *A* take an unlabeled example and label it for *B*?

The answers lie in the estimate of the predictive confidence

Co-training without two sufficient and redundant views [S. Goldman & Y. Zhou, ICML00] (con't)

Combining:

- for each hypothesis and for each equivalence class within the two hypotheses, 10-fold CV on labeled examples is used to compute the 95%-confidence interval [*l*, *h*]
- in making prediction on *x*, compare the $(l + h)/2$ of the confidence intervals:
  - ✓ *A*
  - ✓ *B*
  - ✓ the equivalence class of *A* that contains *x*
  - ✓ the equivalence class of *B* that contains *x*,

  then predict according to the hypothesis that corresponds to the maximum of these four quantities

Co-training without two sufficient and redundant views [S. Goldman & Y. Zhou, ICML00] (con't)

Choosing which examples to label:

Intuitively, classifier *A* should only consider labeling example *x* for **B** if:

- *A*'s confidence in the validity of its label for *x* is better than *B*'s confidence in the validity of its label for *x*

  10-fold CV on labeled examples is used

Weakness of the algorithm:

time-consuming 10-fold CV is used for many times in every round of the co-training process

Tri-training

[Z.-H. Zhou & M. Li, TKDE05]

In order to determine which unlabeled example to label and how to combine the co-trained classifiers, the labeling confidence should be measured explicitly

If three classifiers are involved, maybe it is not necessary to measure the labeling confidence explicitly

- ➢ if two classifiers agree, then label for the other classifier
- ➢ the prediction can be made by voting these three classifiers

Additional benefits:

- Ensemble learning can be utilized to improve the generalization
- Easy to be coupled with a popular active learning scheme, *query-by-committee*

Tri-training (con't)

[Z.-H. Zhou & M. Li, TKDE05]

Since tri-training does not require sufficient and redundant views, nor does it require the use of different supervised learning algorithms whose hypothesis partitions the instance space into a set of equivalence classes, the diversity among the classifiers have to be sought from other channels

Bagging can do

- If the prediction of h_2 and h_3 on x is correct,
  then h_1 will receive a valid new example for further training
- Otherwise,
  h_1 will get an example with noisy label
  however, even in the worse case, the increase in the classification noise rate can be compensated if the amount of newly labeled examples is sufficient, under certain conditions

Tri-training (con’t)

[Z.-H. Zhou & M. Li, TKDE05]

According to [D. Angluin & P. Laird, MLJ88], if a sequence σ of m samples is drawn, where the sample size m satisfies

$$m \geq \frac{2}{\epsilon^2 (1 - 2\eta)^2} \ln \left(\frac{2N}{\delta} \right)$$

ε : the hypothesis worst-case classification error rate
η (< 0.5): an upper bound on the classification noise rate
N: the number of hypothesis
δ: the confidence

then a hypothesis H_i that minimizes disagreement with σ will have the PAC property:

$$\Pr \left[d(H_i, H^*) \geq \epsilon \right] \leq \delta$$

d(,): the sum over the probability of elements from the symmetric difference between the two hypothesis sets H_i and H^* (the ground-truth)

Tri-training (con't)

[Z.-H. Zhou & M. Li, TKDE05]

Let $c = 2\mu \ln\left(\frac{2N}{\delta}\right)$ where μ makes $m \geq \frac{2}{\epsilon^2 (1-2\eta)^2} \ln\left(\frac{2N}{\delta}\right)$ hold equality:

$$m = \frac{c}{\epsilon^2 (1-2\eta)^2}$$

Let $u = \frac{c}{\epsilon^2} = m(1-2\eta)^2$, then $u^t > u^{t\text{-}1}$ implies $\varepsilon^t < \varepsilon^{t\text{-}1}$

The classification noise rate of the training set for h_1 in the t-th round is:

$$\eta^t = \frac{\eta_L |L| + \breve{e}_1^t |L^t|}{|L \cup L^t|}$$

L: the original labeled example set

η_L: the classification noise rate of L

L^t: the set of unlabeled examples labeled by h_2&h_3 for h_1 in the t-th round

$\breve{e}_1^t$: the upper bound of the classification error rate of h_2&h_3 in the t-th round

Tri-training (con't)

[Z.-H. Zhou & M. Li, TKDE05]

Thus,

$$u^t = m^t\left(1 - 2\eta^t\right)^2 = \left|L \cup L^t\right|\left(1 - 2\frac{\eta_L\left|L\right| + \breve{e}_1^t\left|L^t\right|}{\left|L \cup L^t\right|}\right)^2$$

$$u^{t-1} = m^{t-1}\left(1 - 2\eta^{t-1}\right)^2 = \left|L \cup L^{t-1}\right|\left(1 - 2\frac{\eta_L\left|L\right| + \breve{e}_1^{t-1}\left|L^{t-1}\right|}{\left|L \cup L^{t-1}\right|}\right)^2$$

Since $u^t > u^{t\text{-}1}$ implies $\varepsilon^t < \varepsilon^{t\text{-}1}$, h_1 can be improved through using L^t in its training if

$$\left|L \cup L^t\right|\left(1 - 2\frac{\eta_L\left|L\right| + \breve{e}_1^t\left|L^t\right|}{\left|L \cup L^t\right|}\right)^2 > \left|L \cup L^{t-1}\right|\left(1 - 2\frac{\eta_L\left|L\right| + \breve{e}_1^{t-1}\left|L^{t-1}\right|}{\left|L \cup L^{t-1}\right|}\right)^2$$

Considering that η_L can be very small and assume $0 \le \breve{e}_1^t, \breve{e}_1^{t-1} < 0.5$,

the tri-training criterion: $0 < \dfrac{\breve{e}_1^t}{\breve{e}_1^{t-1}} < \dfrac{\left|L^{t-1}\right|}{\left|L^t\right|} < 1$

Tri-training (con’t)

[Z.-H. Zhou & M. Li, TKDE05]

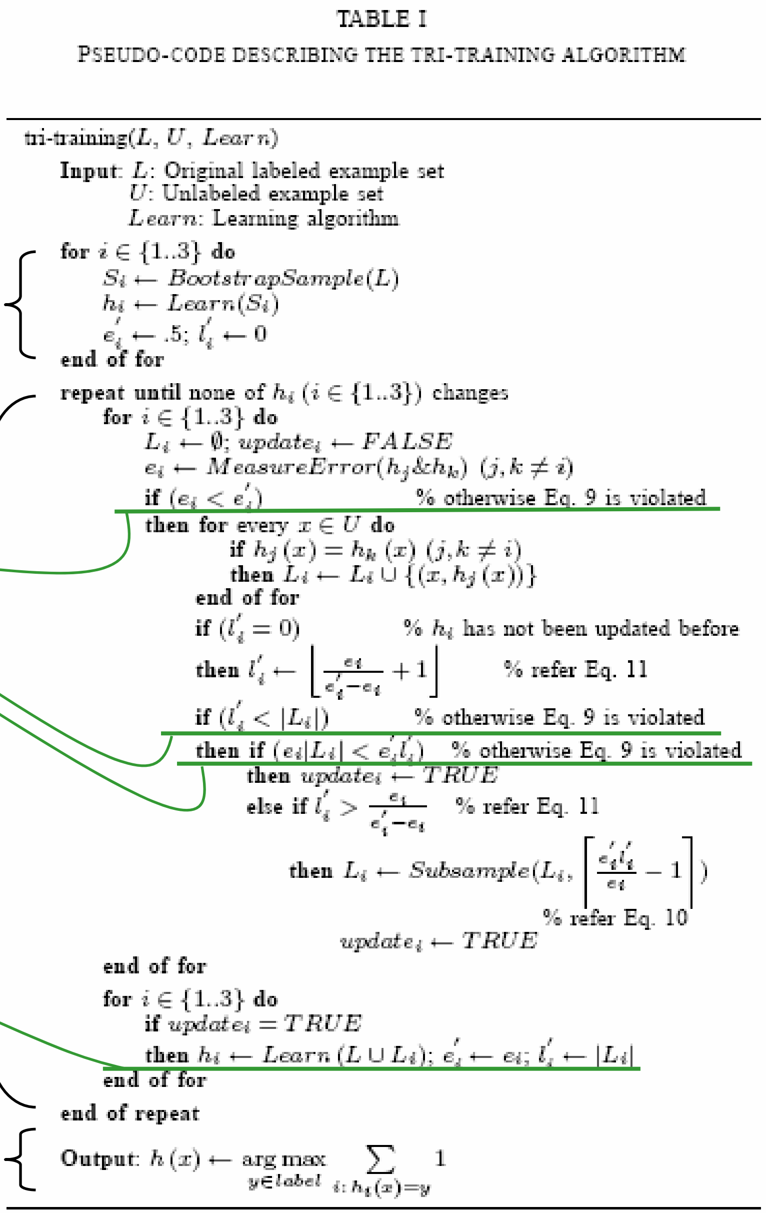


TABLE I

PSEUDO-CODE DESCRIBING THE TRI-TRAINING ALGORITHM

```
tri-training($L$, $U$, $Learn$)
  Input: $L$: Original labeled example set
         $U$: Unlabeled example set
         $Learn$: Learning algorithm
  for $i \in \{1..3\}$ do
      $S_i \leftarrow BootstrapSample(L)$
      $h_i \leftarrow Learn(S_i)$
      $e_i' \leftarrow .5$; $l_i' \leftarrow 0$
  end of for
  repeat until none of $h_i$ ($i \in \{1..3\}$) changes
      for $i \in \{1..3\}$ do
          $L_i \leftarrow \emptyset$; $update_i \leftarrow FALSE$
          $e_i \leftarrow MeasureError(h_j \& h_k)$ $(j,k \neq i)$
          if ($e_i < e_i'$)                % otherwise Eq. 9 is violated
          then for every $x \in U$ do
                  if $h_j(x) = h_k(x)$ $(j,k \neq i)$
                  then $L_i \leftarrow L_i \cup \{(x, h_j(x))\}$
               end of for
               if ($l_i' = 0$)             % $h_i$ has not been updated before
               then $l_i' \leftarrow \left\lfloor \frac{e_i}{e_i' - e_i} + 1 \right\rfloor$        % refer Eq. 11
               if ($l_i' < |L_i|$)          % otherwise Eq. 9 is violated
               then if ($e_i|L_i| < e_i' l_i'$)   % otherwise Eq. 9 is violated
                    then $update_i \leftarrow TRUE$
                    else if $l_i' > \frac{e_i}{e_i' - e_i}$    % refer Eq. 11
                         then $L_i \leftarrow Subsample(L_i, \left\lceil \frac{e_i' l_i'}{e_i} - 1 \right\rceil)$
                                                        % refer Eq. 10
                              $update_i \leftarrow TRUE$
      end of for
      for $i \in \{1..3\}$ do
          if $update_i = TRUE$
          then $h_i \leftarrow Learn(L \cup L_i)$; $e_i' \leftarrow e_i$; $l_i' \leftarrow |L_i|$
      end of for
  end of repeat
  Output: $h(x) \leftarrow \arg\max_{y \in label} \sum_{i: h_i(x) = y} 1$
```

Use Bagging to generate the initial (3) classifiers

Check whether the tri-training criterion satisfies

Tri-training

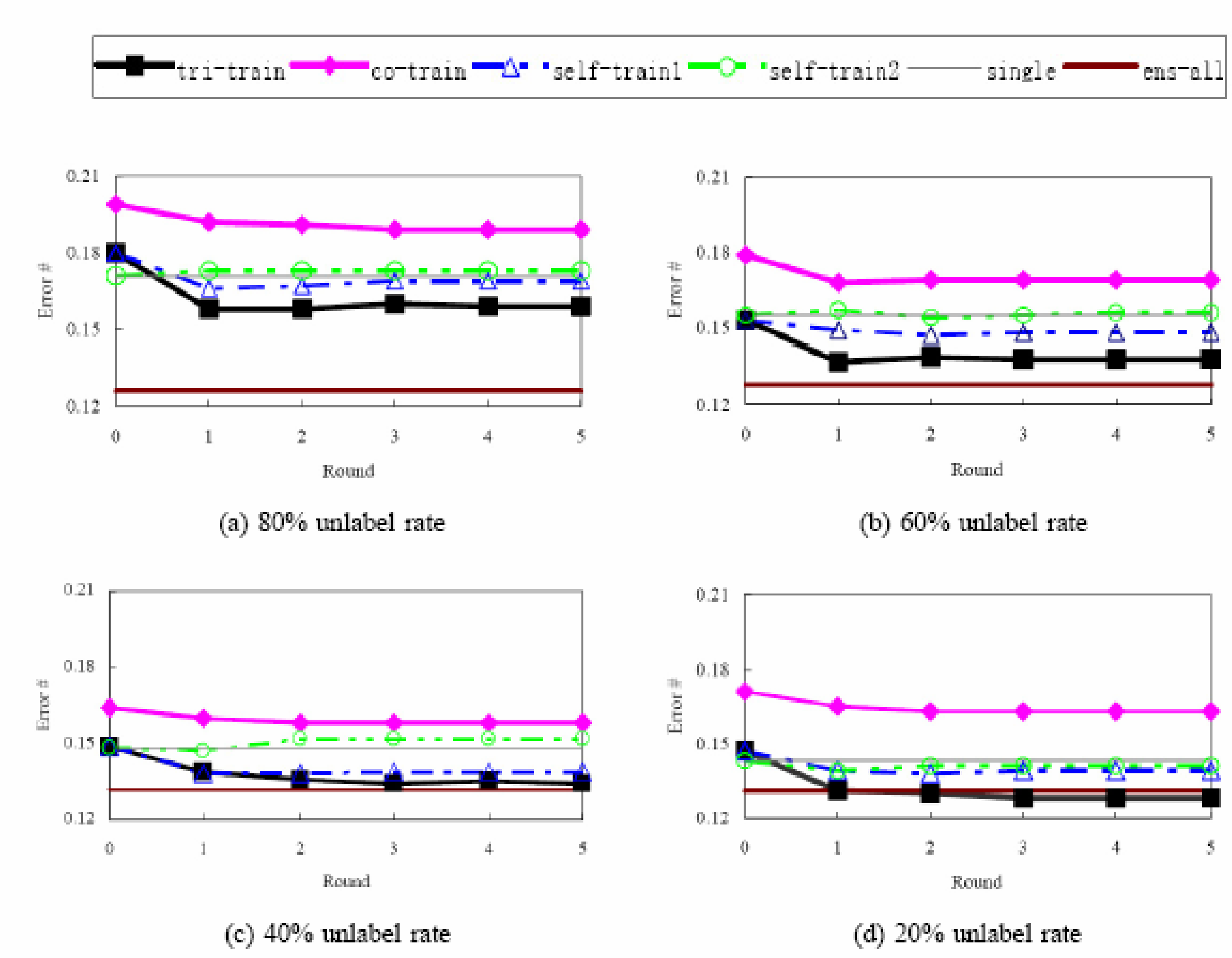
refine the classifiers

Vote the (3) classifiers in prediction

Tri-training (con't)

[Z.-H. Zhou & M. Li, TKDE05]

On UCI datasets



(a) 80% unlabel rate

(b) 60% unlabel rate

(c) 40% unlabel rate

(d) 20% unlabel rate

Fig. 1. Error rates averaged across all the data sets when J4.8 decision trees are used

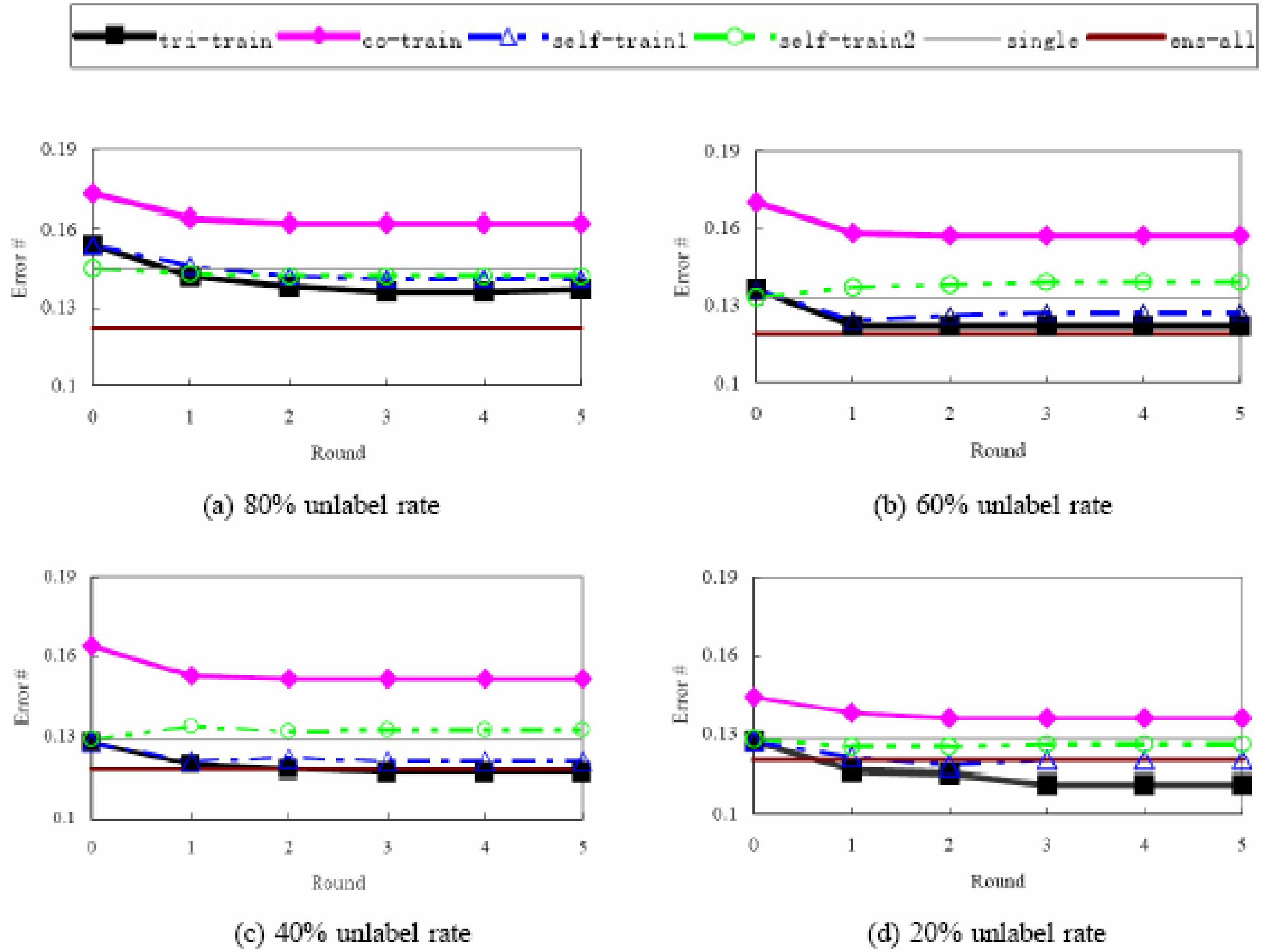
Tri-training (con't)

[Z.-H. Zhou & M. Li, TKDE05]

Learning And Mining from DatA

http://lamda.nju.edu.cn

On UCI datasets



(a) 80% unlabel rate

(b) 60% unlabel rate

(c) 40% unlabel rate

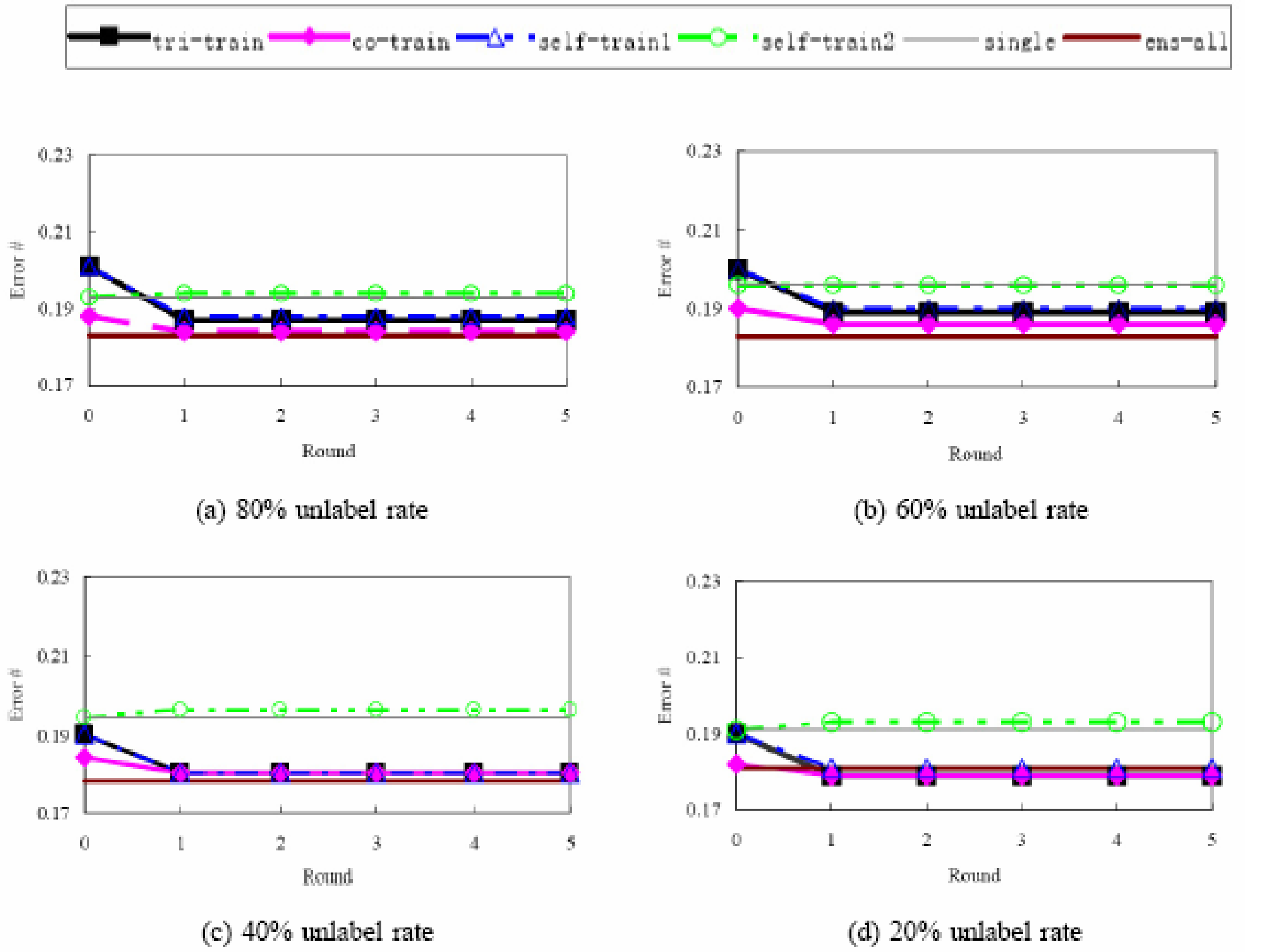
(d) 20% unlabel rate

Fig. 2. Error rates averaged across all the data sets when BP neural networks are used

Tri-training (con't)

[Z.-H. Zhou & M. Li, TKDE05]

On UCI datasets



(a) 80% unlabel rate

(b) 60% unlabel rate

(c) 40% unlabel rate

(d) 20% unlabel rate

Fig. 3. Error rates averaged across all the data sets when Naive Bayes classifiers are used

Tri-training (con’t)

[Z.-H. Zhou & M. Li, TKDE05]

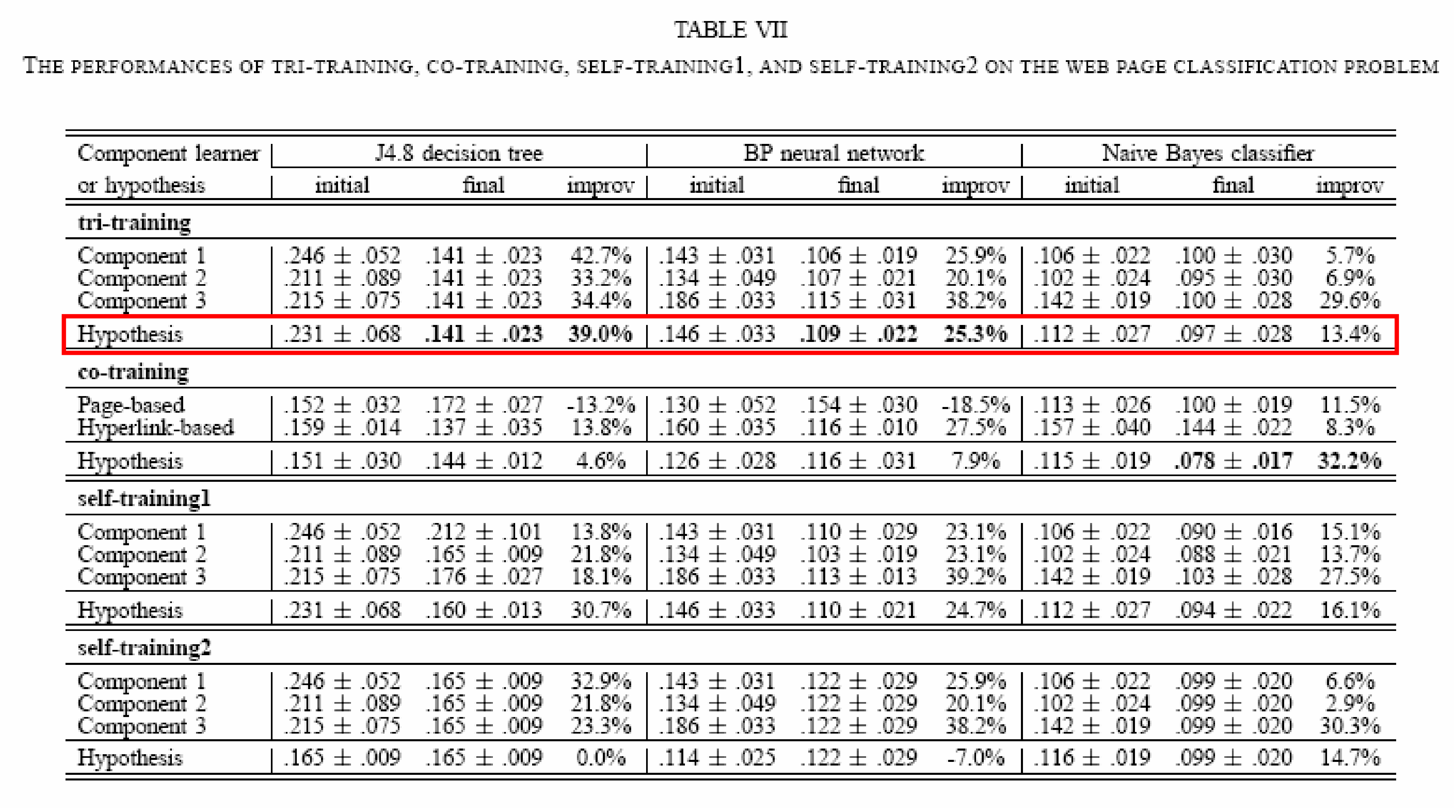


TABLE VII

THE PERFORMANCES OF TRI-TRAINING, CO-TRAINING, SELF-TRAINING1, AND SELF-TRAINING2 ON THE WEB PAGE CLASSIFICATION PROBLEM

Component learner	J4.8 decision tree			BP neural network			Naive Bayes classifier		
or hypothesis	initial	final	improv	initial	final	improv	initial	final	improv
tri-training									
Component 1	.246 ± .052	.141 ± .023	42.7%	.143 ± .031	.106 ± .019	25.9%	.106 ± .022	.100 ± .030	5.7%
Component 2	.211 ± .089	.141 ± .023	33.2%	.134 ± .049	.107 ± .021	20.1%	.102 ± .024	.095 ± .030	6.9%
Component 3	.215 ± .075	.141 ± .023	34.4%	.186 ± .033	.115 ± .031	38.2%	.142 ± .019	.100 ± .028	29.6%
Hypothesis	.231 ± .068	**.141 ± .023**	**39.0%**	.146 ± .033	**.109 ± .022**	**25.3%**	.112 ± .027	.097 ± .028	13.4%
co-training									
Page-based	.152 ± .032	.172 ± .027	-13.2%	.130 ± .052	.154 ± .030	-18.5%	.113 ± .026	.100 ± .019	11.5%
Hyperlink-based	.159 ± .014	.137 ± .035	13.8%	.160 ± .035	.116 ± .010	27.5%	.157 ± .040	.144 ± .022	8.3%
Hypothesis	.151 ± .030	.144 ± .012	4.6%	.126 ± .028	.116 ± .031	7.9%	.115 ± .019	**.078 ± .017**	**32.2%**
self-training1									
Component 1	.246 ± .052	.212 ± .101	13.8%	.143 ± .031	.110 ± .029	23.1%	.106 ± .022	.090 ± .016	15.1%
Component 2	.211 ± .089	.165 ± .009	21.8%	.134 ± .049	.103 ± .019	23.1%	.102 ± .024	.088 ± .021	13.7%
Component 3	.215 ± .075	.176 ± .027	18.1%	.186 ± .033	.113 ± .013	39.2%	.142 ± .019	.103 ± .028	27.5%
Hypothesis	.231 ± .068	.160 ± .013	30.7%	.146 ± .033	.110 ± .021	24.7%	.112 ± .027	.094 ± .022	16.1%
self-training2									
Component 1	.246 ± .052	.165 ± .009	32.9%	.143 ± .031	.122 ± .029	25.9%	.106 ± .022	.099 ± .020	6.6%
Component 2	.211 ± .089	.165 ± .009	21.8%	.134 ± .049	.122 ± .029	20.1%	.102 ± .024	.099 ± .020	2.9%
Component 3	.215 ± .075	.165 ± .009	23.3%	.186 ± .033	.122 ± .029	38.2%	.142 ± .019	.099 ± .020	30.3%
Hypothesis	.165 ± .009	.165 ± .009	0.0%	.114 ± .025	.122 ± .029	-7.0%	.116 ± .019	.099 ± .020	14.7%

Semi-supervised regression with co-training

[Z.-H. Zhou & M. Li, IJCAI05]

Previous research on semi-supervised learning mainly focuses on classification (where the prediction is discrete variables)

Up to our knowledge, this is the first work on semi-supervised regression (where the prediction is continuous variables)

COREG (CO-training REGressors)

This is a co-training style algorithm

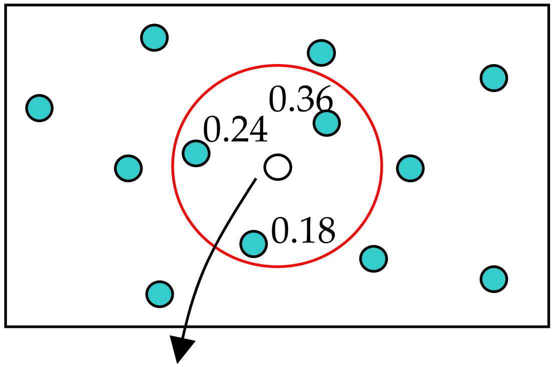
The key ideas of COREG can also be used to develop other kinds of semi-supervised regression algorithms

Semi-supervised regression with co-training

[Z.-H. Zhou & M. Li, IJCAI05] (con't)

kNN regressor is used as the base learner to instantiate the two regressors:

a new instance is labeled through averaging the real-valued labels of its *k*-nearest neighboring examples



Prediction: 0.26 = (0.36 + 0.24 + 0.18) /3

- Easy to be refined in each of the learning iterations

  If neural networks or regression trees were used, then in each iteration the regressors have to be re-trained with the labeled examples in addition to the newly labeled ones

- Easy to be coupled with the process for estimating the labeling confidence

  The process for estimating the labeling confidence in COREG utilizes the neighboring properties of the training examples

Semi-supervised regression with co-training

[Z.-H. Zhou & M. Li, IJCAI05] (con't)

COREG does not assume sufficient and redundant views

The initial regressors should be diverse

If they are identical, then for either regressor, the unlabeled examples labeled by the other regressor may be the same as these labeled by the regressor for itself

The solution: Use different distance metrics

$$Minkowsky_p(\mathbf{x}_r, \mathbf{x}_s) = \left(\sum_{l=1}^{d} |\mathbf{x}_{r,l} - \mathbf{x}_{s,l}|^p \right)^{1/p}$$

Different values of *p* result in different vicinities identified

- ➢ smaller *p* value → more robust to data variations
- ➢ bigger *p* value → more sensitive to data variations

Additional benefit: it is usually difficult to determine which *p* value is better, therefore the functions of the regressors may be somewhat complementary to be combined

Semi-supervised regression with co-training [Z.-H. Zhou & M. Li, IJCAI05] (con't)

In classification, estimating the labeling confidence is straightforward because besides the class label, many classifiers provide an estimated probability (or an approximation) for the classification

e.g.

- Naïve Bayes classifier returns the maximum posteriori hypothesis where the posterior probabilities can be used
- BP neural network returns the thresholded classification where the real-valued outputs can be used

Two instances:

a: 0.90 (c_1), 0.10 (c_2)
b: 0.60 (c_1), 0.40 (c_2)

⇒

Both a and b are classified to class c_1, but a is more confident to be labeled

Semi-supervised regression with co-training

[Z.-H. Zhou & M. Li, IJCAI05] (con't)

In regression, there is no such estimated probability can be used directly

because in contrast to classification where the number of class labels to be predicted is finite, the possible predictions in regression is infinite

So, a key of COREG is the mechanism for estimating the labeling confidence

Semi-supervised regression with co-training

[Z.-H. Zhou & M. Li, IJCAI05] (con't)

The solution: regarding the labeling of the unlabeled example which makes the regressor most consistent with the labeled example set as with the most confidence

The error of the regressor on the labeled example set should decrease the most if the most confidently labeled example is utilized

Δ: the MSE of the regressor on the labeled example set

Δ′: the MSE of the refined regressor (with $(\mathbf{x}_u, \hat{y}_u)$) on the labeled example set

$\Delta_u = \Delta - \Delta'$

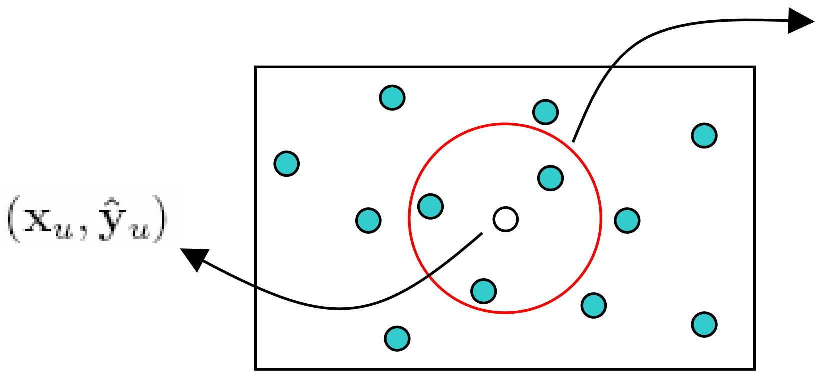
$(\mathbf{x}_u, \hat{y}_u)$ with the biggest positive Δ_u is regarded as the most confidently labeled example

Semi-supervised regression with co-training

[Z.-H. Zhou & M. Li, IJCAI05] (con’t)

Repeatedly measuring the MSE of the *k*NN regressor on the whole labeled example set in each iteration will be time-consuming

An approximation is used, considering that *k*NN mainly utilizes local information



Ω: the vicinity of $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$

h(*x*): the original regressor

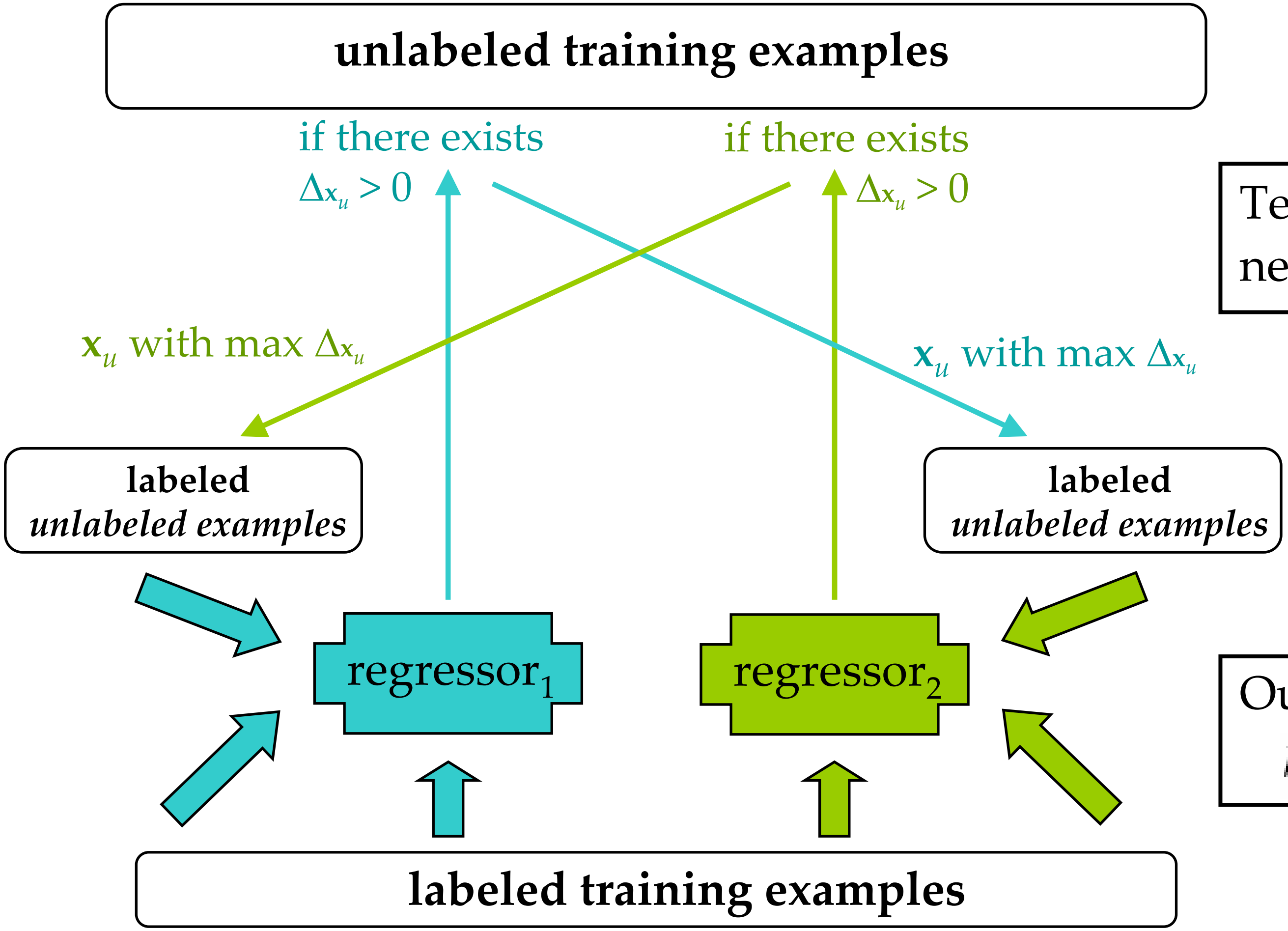
h′(*x*): the refined regressor (with $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$)

$\hat{\mathbf{y}}_u = h(\mathbf{x}_u)$

$$\Delta_{\mathbf{x}_u} = \sum_{\mathbf{x}_i \in \Omega} \left(\underbrace{(\mathbf{y}_i - h(\mathbf{x}_i))^2}_{\Delta \text{ on } \Omega} - \underbrace{(\mathbf{y}_i - h'(\mathbf{x}_i))^2}_{\Delta' \text{ on } \Omega} \right)$$

Semi-supervised regression with co-training

[Z.-H. Zhou & M. Li, IJCAI05] (con’t)



unlabeled training examples

if there exists $\Delta_{\mathbf{x}_u} > 0$

if there exists $\Delta_{\mathbf{x}_u} > 0$

$\mathbf{x}_u$ with max $\Delta_{\mathbf{x}_u}$

$\mathbf{x}_u$ with max $\Delta_{\mathbf{x}_u}$

labeled *unlabeled examples*

labeled *unlabeled examples*

regressor_1

regressor_2

labeled training examples

Terminate:
neither regressor changes

Output:

$$h^*(\mathbf{x}) \leftarrow \frac{1}{2}\left(h_1(\mathbf{x}) + h_2(\mathbf{x})\right)$$

Semi-supervised regression with co-training [Z.-H. Zhou & M. Li, IJCAI05] (con’t)

improv. = (initial MSE – final MSE)/ initial MSE

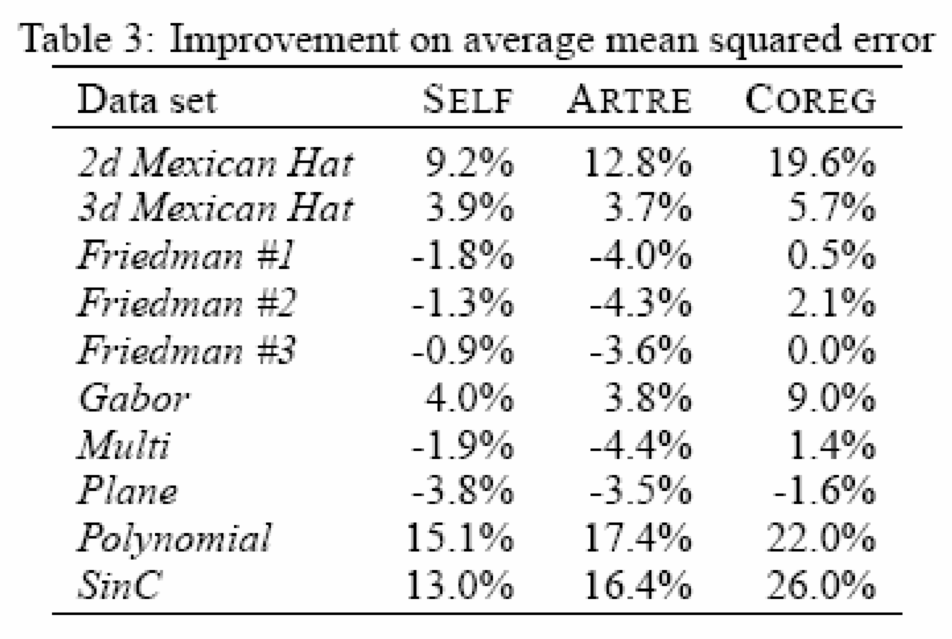


Table 3: Improvement on average mean squared error

Data set	SELF	ARTRE	COREG
2d Mexican Hat	9.2%	12.8%	19.6%
3d Mexican Hat	3.9%	3.7%	5.7%
Friedman #1	-1.8%	-4.0%	0.5%
Friedman #2	-1.3%	-4.3%	2.1%
Friedman #3	-0.9%	-3.6%	0.0%
Gabor	4.0%	3.8%	9.0%
Multi	-1.9%	-4.4%	1.4%
Plane	-3.8%	-3.5%	-1.6%
Polynomial	15.1%	17.4%	22.0%
SinC	13.0%	16.4%	26.0%

- COREG improves on 8 data sets
- SELF/ARTRE improves on 5 data sets
- Improv. of COREG is always bigger than that of SELF/ARTRE

Table reprinted from [Z.-H. Zhou & M. Li, IJCAI05]

Self-training with editing

[M. Li & Z.-H. Zhou, PAKDD05]

In the process of semi-supervised learning, the negative effect caused by mislabeled *unlabeled examples* will be accumulated, which will seriously degrade the performance

> *Corrected co-training* [D. Pierce & C. Cardie, EMNLP01; R. Hwa et al., ICML03w]: to have a human intervene by reviewing and correcting the instances labeled by the view classifiers

Is it possible to use data editing methods to filter out the mislabeled examples?

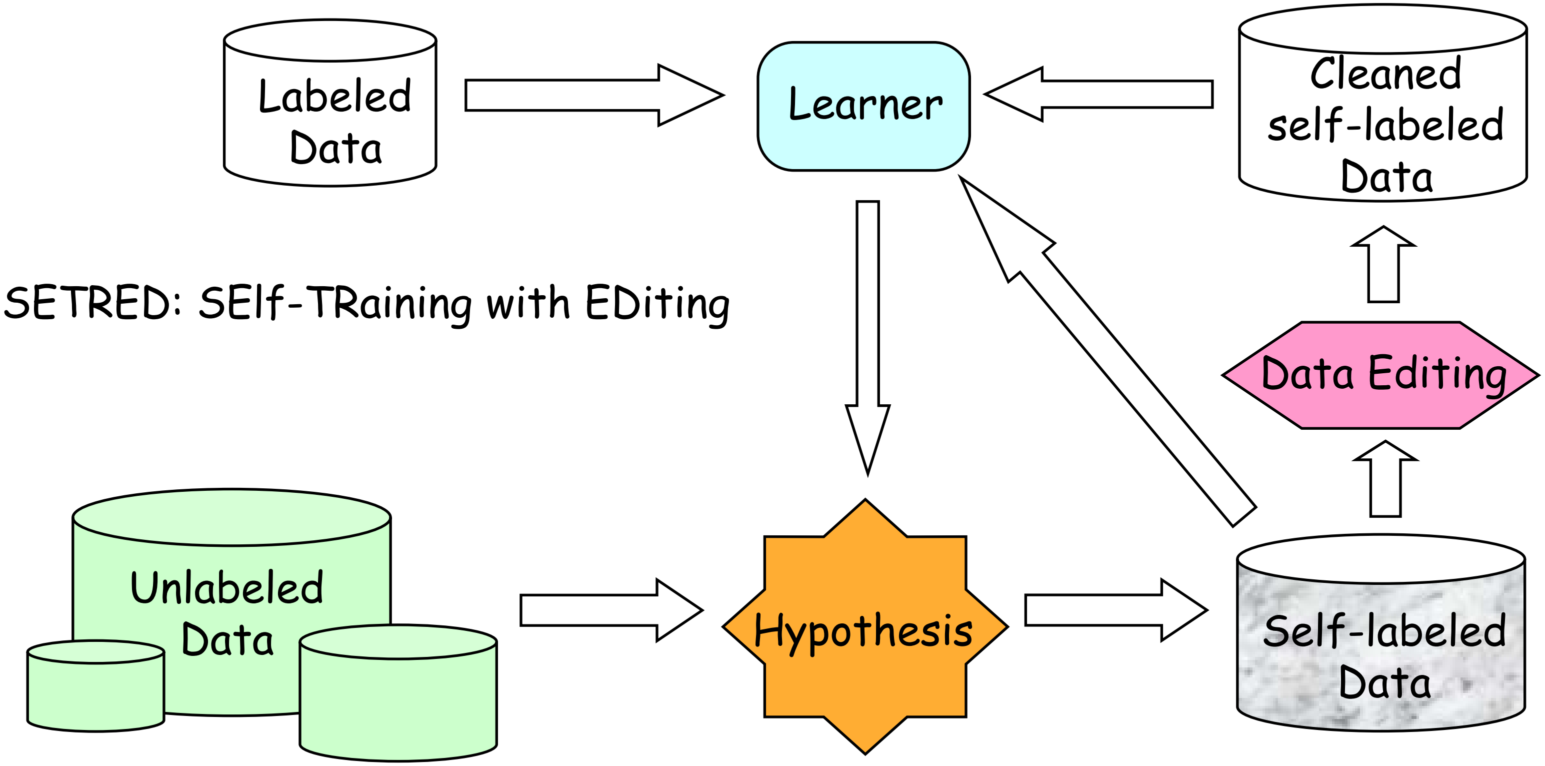
Data editing: a technique which attempts to improve the quality of the training set through identifying and eliminating the training examples wrongly generated in the human labeling process

> Some effective methods: [D.R. Wilson, TSMC72; J. Koplowitz & T.A. Brown, PR81; J.S. Sánchez et al., PRL03; Y. Jiang & Z.-H. Zhou, ISNN05]

Self-training with editing

[M. Li & Z.-H. Zhou, PAKDD05] (con't)

The self-training algorithm [K. Nigam & R. Ghani, CIKM00] seriously suffers from mislabeled unlabeled examples

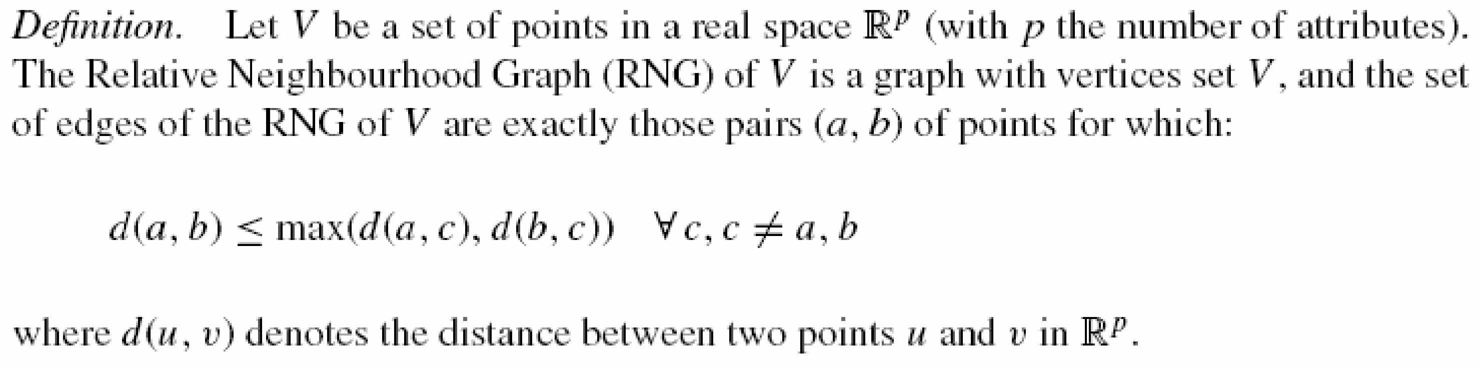


SETRED: SElf-TRaining with EDiting

Self-training with editing

[M. Li & Z.-H. Zhou, PAKDD05] (con't)

The data editing method used in SETRED is a method based on *Relative Neighborhood Graph (RNG)* and *cut edge weight* statistic [F. Muhlenbach et al., JIIS04]



Definition. Let V be a set of points in a real space $\mathbb{R}^p$ (with p the number of attributes). The Relative Neighbourhood Graph (RNG) of V is a graph with vertices set V, and the set of edges of the RNG of V are exactly those pairs (a, b) of points for which:

$$d(a,b) \leq \max(d(a,c), d(b,c)) \quad \forall c, c \neq a, b$$

where $d(u, v)$ denotes the distance between two points u and v in $\mathbb{R}^p$.

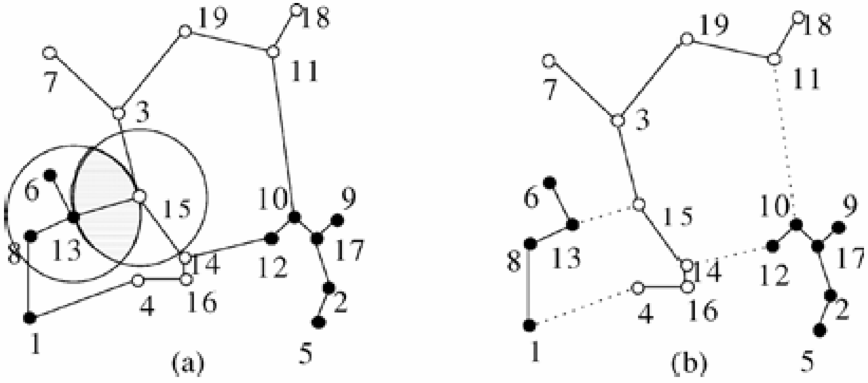


Figure 1. Relative Neighbourhood Graph and Clusters with two Classes: the black and the white points.

An edge connecting two vertices that have different labels is called *cut edge*

Figure reprinted from [F. Muhlenbach et al., JIIS04]

Self-training with editing

[M. Li & Z.-H. Zhou, PAKDD05] (con’t)

UCI datasets, 25% test, 75% training (*unlabel rate* 90%)

Nearest Neighbor (NN) Classifiers are used

Baselines: NN-L: NN trained from *L* only

NN-A: NN trained from *L*∪*N* with all labels known

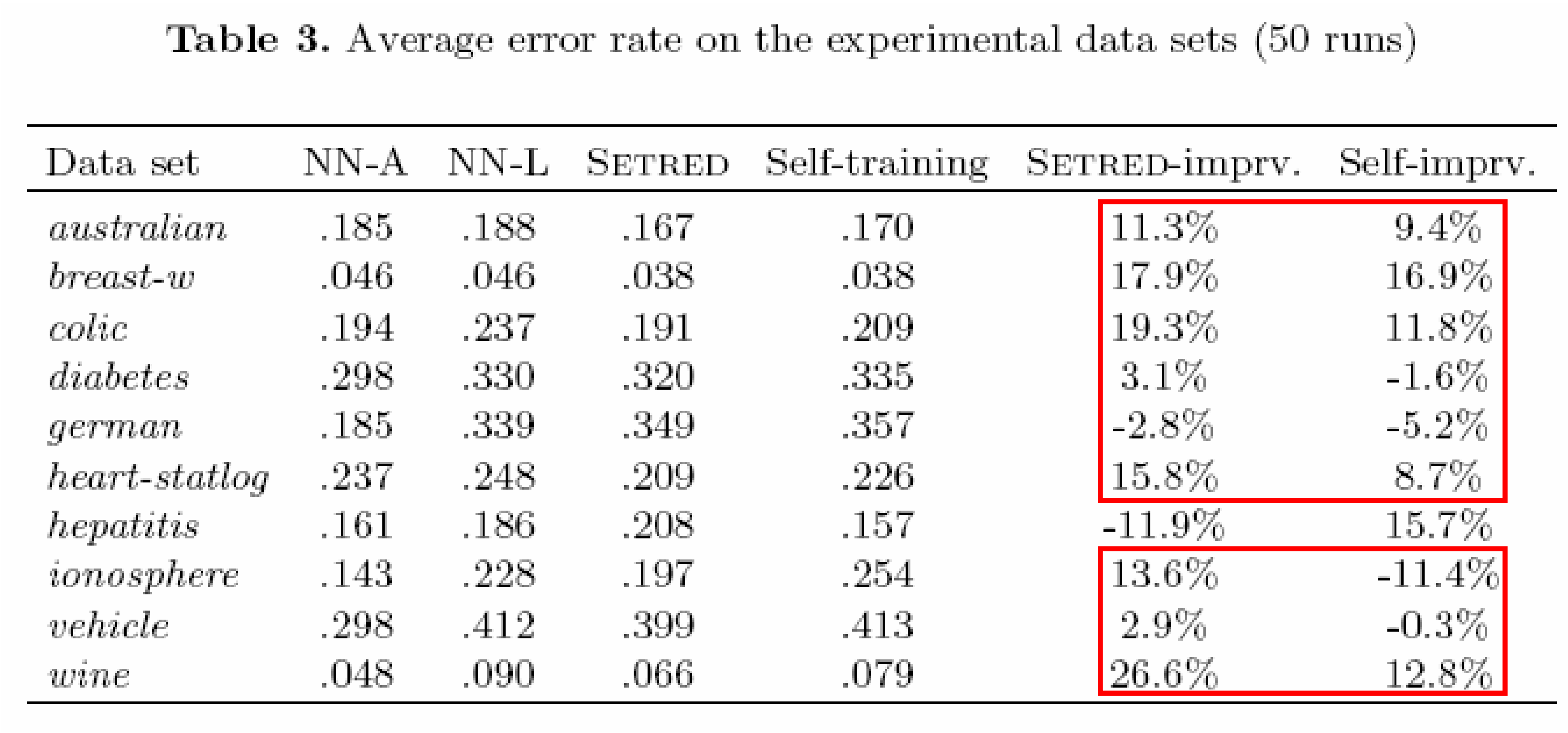


Table 3. Average error rate on the experimental data sets (50 runs)

Data set	NN-A	NN-L	SETRED	Self-training	SETRED-imprv.	Self-imprv.
australian	.185	.188	.167	.170	11.3%	9.4%
breast-w	.046	.046	.038	.038	17.9%	16.9%
colic	.194	.237	.191	.209	19.3%	11.8%
diabetes	.298	.330	.320	.335	3.1%	-1.6%
german	.185	.339	.349	.357	-2.8%	-5.2%
heart-statlog	.237	.248	.209	.226	15.8%	8.7%
hepatitis	.161	.186	.208	.157	-11.9%	15.7%
ionosphere	.143	.228	.197	.254	13.6%	-11.4%
vehicle	.298	.412	.399	.413	2.9%	-0.3%
wine	.048	.090	.066	.079	26.6%	12.8%

Table reprinted from [M. Li & Z.-H. Zhou, PAKDD05]

Future extensions

- Extend tri-training to include more learners
- Incorporate query-by-committee in tri-training
- Effective data editing schemes for semi-supervised learning
- Generate diverse initial learners in tri-training and COREG
- Designing semi-supervised regression algorithms based on the key idea of COREG
- … …

Thanks!