Chapter 3: Combining Classifiers

From "*Web Data Mining*", by Bing Liu (UIC), Springer Verlag, 2007

Outline

- Ensemble methods: Bagging and Boosting
- Fully supervised learning (traditional classification)
- Partially (semi-) supervised learning (or classification)
  - Learning with a small set of labeled examples and a large set of unlabeled examples (LU learning)

Combining classifiers

- So far, we have only discussed individual classifiers, i.e., how to build them and use them.
- Can we combine multiple classifiers to produce a better classifier?
- Yes, sometimes
- We discuss two main algorithms:
  - Bagging
  - Boosting

Bagging

- Breiman, 1996
- Bootstrap Aggregating = Bagging
  - Application of bootstrap sampling
    - Given: set *D* containing *m* training examples
    - Create a sample *S*[*i*] of *D* by drawing *m* examples at random *with replacement* from *D*
    - *S*[*i*] of size *m*: expected to leave out 0.37 of examples from *D*

Bagging (cont…)

- Training
  - Create *k* bootstrap samples *S*[1], *S*[2], …, *S*[*k*]
  - Build a distinct classifier on each *S*[*i*] to produce *k* classifiers, using the same learning algorithm.
- Testing
  - Classify each new instance by voting of the *k* classifiers (equal weights)

Bagging Example

Original	1	2	3	4	5	6	7	8
Training set 1	2	7	8	3	7	6	3	1
Training set 2	7	8	5	6	4	2	7	1
Training set 3	3	6	2	7	5	6	2	2
Training set 4	4	5	1	4	6	4	3	8

Bagging (cont …)

- When does it help?
  - When learner is <u>unstable</u>
    - Small change to training set causes large change in the output classifier
    - True for decision trees, neural networks; not true for *k*-nearest neighbor, naïve Bayesian, class association rules
  - Experimentally, bagging can help substantially for unstable learners, may somewhat degrade results for stable learners

Bagging Predictors, Leo Breiman, 1996

Boosting

- A family of methods:
    - We only study **AdaBoost** (Freund & Schapire, 1996)
- Training
    - Produce a sequence of classifiers (the same base learner)
    - Each classifier is dependent on the previous one, and focuses on the previous one's errors
    - Examples that are incorrectly predicted in previous classifiers are given higher weights
- Testing
    - For a test case, the results of the series of classifiers are combined to determine the final class of the test case.

AdaBoost

Weighted
training set

(x_1, y_1, w_1)
(x_2, y_2, w_2)
…
(x_n, y_n, w_n)

Non-negative weights sum to 1

called a weaker classifier

- Build a classifier ***h*$_t$** whose accuracy on training set > ½ (better than random)

Change weights

AdaBoost algorithm

```
**Algorithm AdaBoost.M1**
**Input:** sequence of $m$ examples $\langle (x_1, y_1), \ldots, (x_m, y_m) \rangle$
           with labels $y_i \in Y = \{1, \ldots, k\}$
       weak learning algorithm **WeakLearn**
       integer $T$ specifying number of iterations
**Initialize** $D_1(i) = 1/m$ for all $i$.
**Do for** $t = 1, 2, \ldots, T$:
    1. Call **WeakLearn**, providing it with the distribution $D_t$.
    2. Get back a hypothesis $h_t : X \to Y$.
    3. Calculate the error of $h_t$: $\epsilon_t = \sum_{i: h_t(x_i) \neq y_i} D_t(i)$.
       If $\epsilon_t > 1/2$, then set $T = t - 1$ and abort loop.
    4. Set $\beta_t = \epsilon_t / (1 - \epsilon_t)$.
    5. Update distribution $D_t$:
       $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t & \text{if } h_t(x_i) = y_i \\ 1 & \text{otherwise} \end{cases}$
       where $Z_t$ is a normalization constant (chosen so that $D_{t+1}$
       will be a distribution).
**Output** the final hypothesis:
    $h_{fin}(x) = \arg\max_{y \in Y} \sum_{t: h_t(x) = y} \log \frac{1}{\beta_t}$.
```

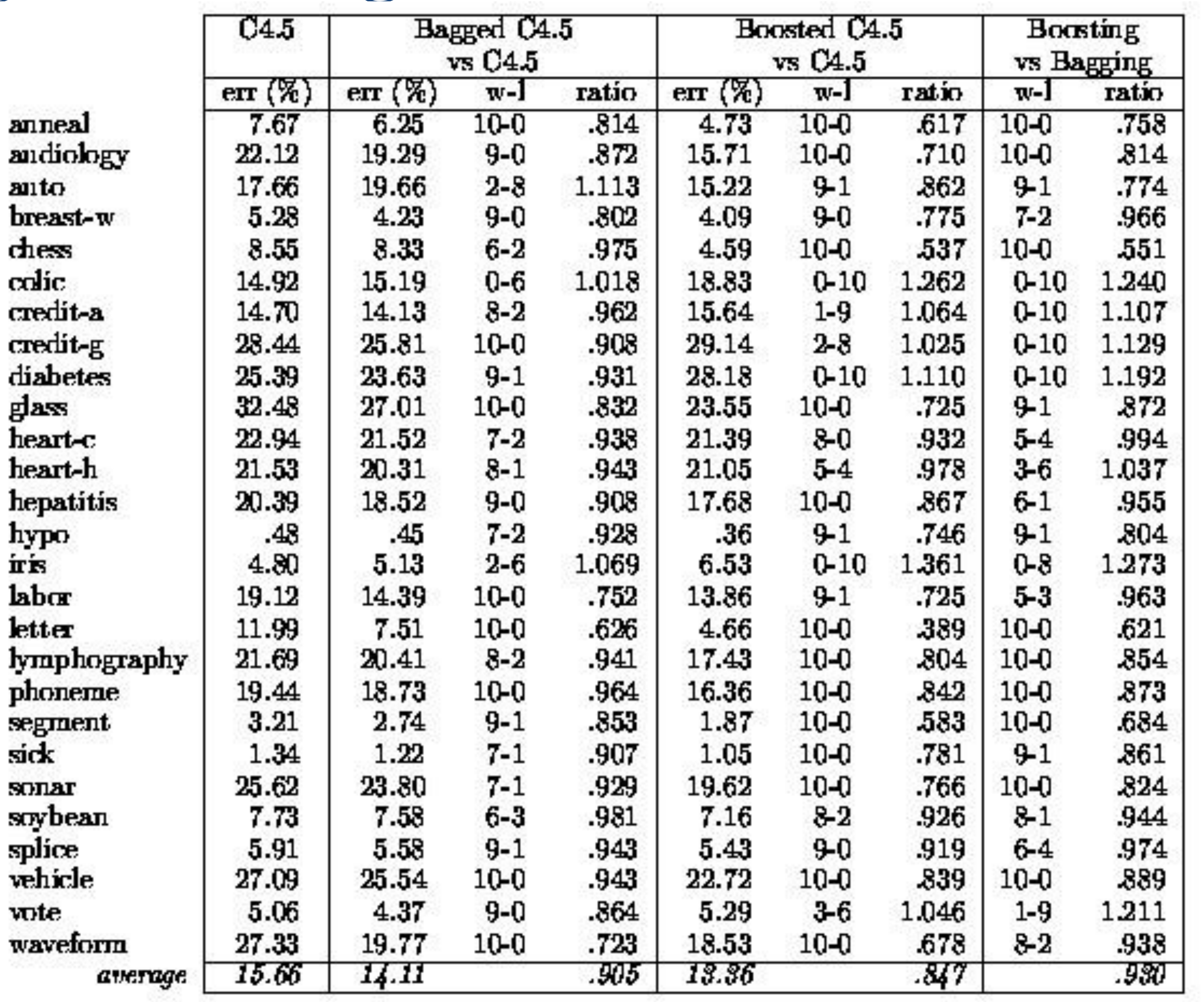
Bagging, Boosting and C4.5

<u>C4.5</u>’s mean error rate over the 10 cross-validation.

<u>Bagged C4.5</u> vs. C4.5.

<u>Boosted C4.5</u> vs. C4.5.

<u>Boosting vs. Bagging</u>

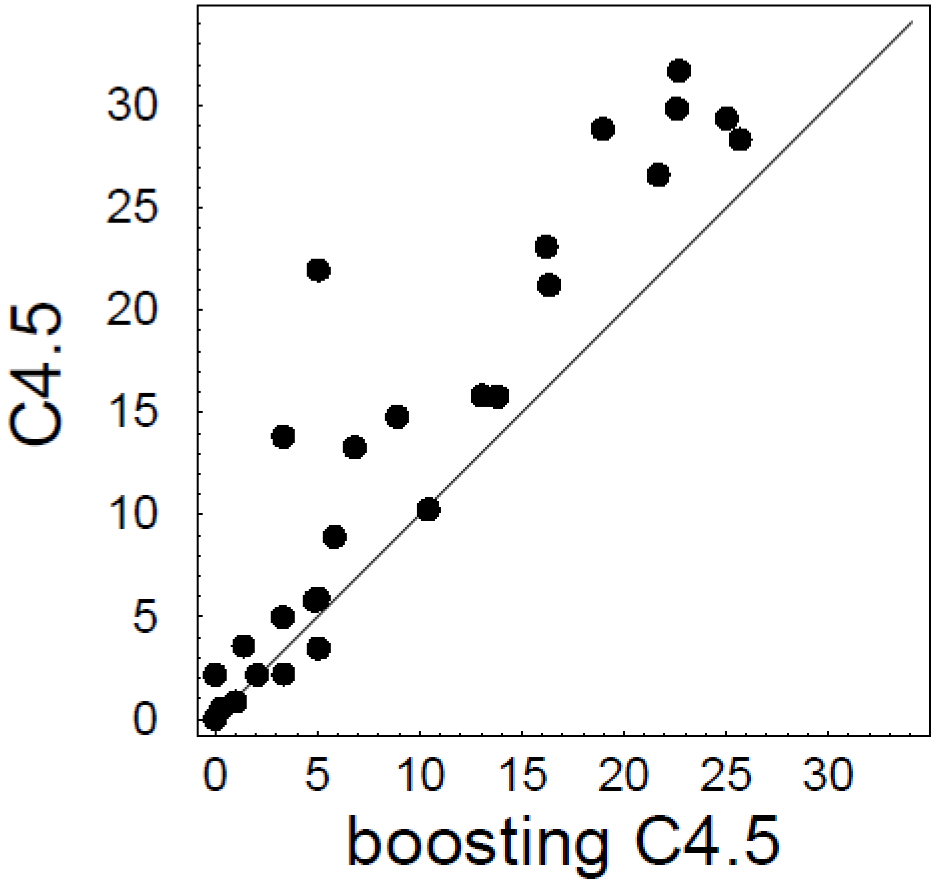


	C4.5	Bagged C4.5 vs C4.5			Boosted C4.5 vs C4.5			Boosting vs Bagging	
	err (%)	err (%)	w-l	ratio	err (%)	w-l	ratio	w-l	ratio
anneal	7.67	6.25	10-0	.814	4.73	10-0	.617	10-0	.758
audiology	22.12	19.29	9-0	.872	15.71	10-0	.710	10-0	.814
auto	17.66	19.66	2-8	1.113	15.22	9-1	.862	9-1	.774
breast-w	5.28	4.23	9-0	.802	4.09	9-0	.775	7-2	.966
chess	8.55	8.33	6-2	.975	4.59	10-0	.537	10-0	.551
colic	14.92	15.19	0-6	1.018	18.83	0-10	1.262	0-10	1.240
credit-a	14.70	14.13	8-2	.962	15.64	1-9	1.064	0-10	1.107
credit-g	28.44	25.81	10-0	.908	29.14	2-8	1.025	0-10	1.129
diabetes	25.39	23.63	9-1	.931	28.18	0-10	1.110	0-10	1.192
glass	32.48	27.01	10-0	.832	23.55	10-0	.725	9-1	.872
heart-c	22.94	21.52	7-2	.938	21.39	8-0	.932	5-4	.994
heart-h	21.53	20.31	8-1	.943	21.05	5-4	.978	3-6	1.037
hepatitis	20.39	18.52	9-0	.908	17.68	10-0	.867	6-1	.955
hypo	.48	.45	7-2	.928	.36	9-1	.746	9-1	.804
iris	4.80	5.13	2-6	1.069	6.53	0-10	1.361	0-8	1.273
labor	19.12	14.39	10-0	.752	13.86	9-1	.725	5-3	.963
letter	11.99	7.51	10-0	.626	4.66	10-0	.389	10-0	.621
lymphography	21.69	20.41	8-2	.941	17.43	10-0	.804	10-0	.854
phoneme	19.44	18.73	10-0	.964	16.36	10-0	.842	10-0	.873
segment	3.21	2.74	9-1	.853	1.87	10-0	.583	10-0	.684
sick	1.34	1.22	7-1	.907	1.05	10-0	.781	9-1	.861
sonar	25.62	23.80	7-1	.929	19.62	10-0	.766	10-0	.824
soybean	7.73	7.58	6-3	.981	7.16	8-2	.926	8-1	.944
splice	5.91	5.58	9-1	.943	5.43	9-0	.919	6-4	.974
vehicle	27.09	25.54	10-0	.943	22.72	10-0	.839	10-0	.889
vote	5.06	4.37	9-0	.864	5.29	3-6	1.046	1-9	1.211
waveform	27.33	19.77	10-0	.723	18.53	10-0	.678	8-2	.938
average	15.66	14.11		.905	13.36		.847		.930

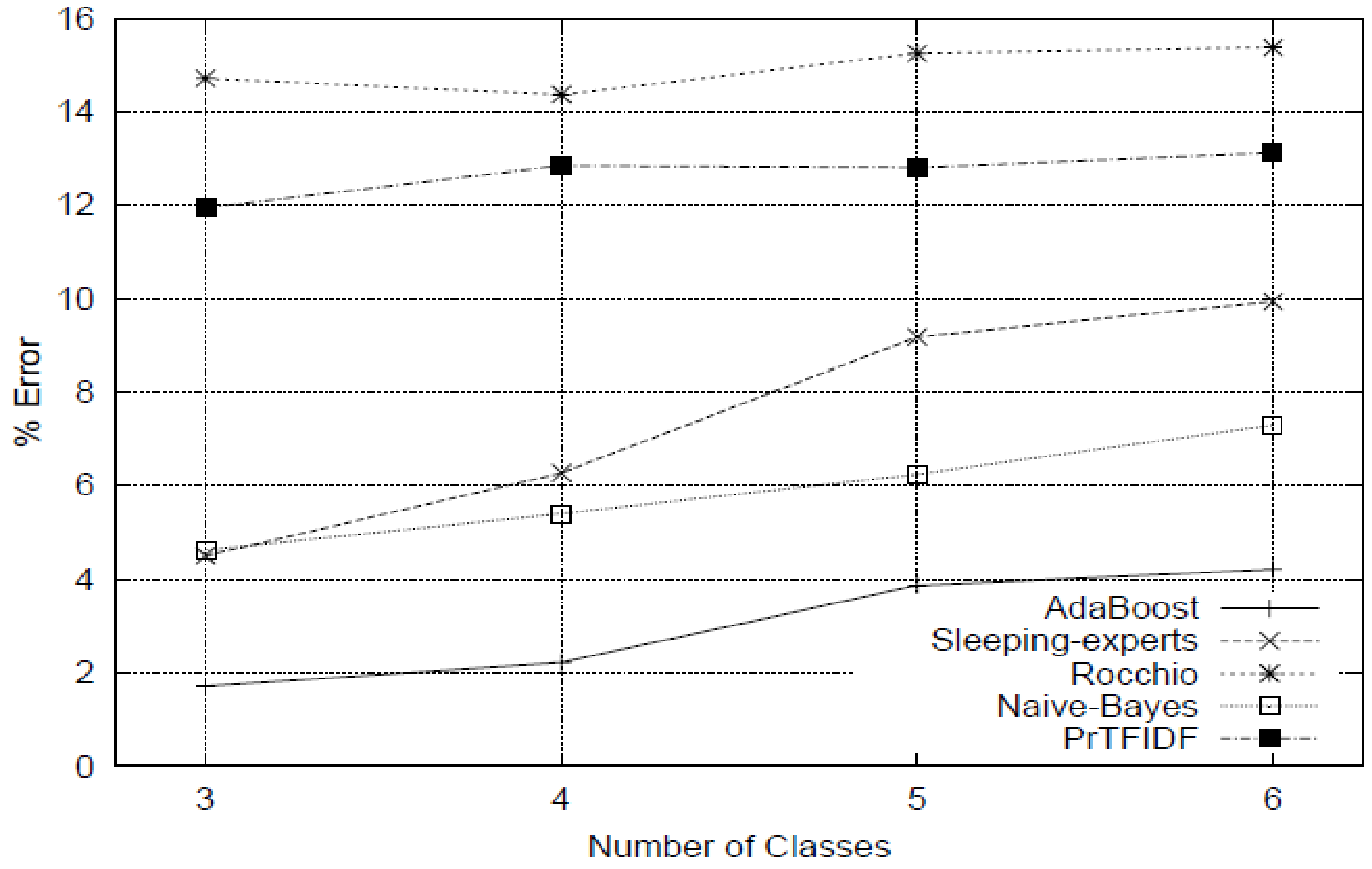
Does AdaBoost always work?

- The actual performance of boosting depends on the data and the base learner.
  - It requires the base learner to be unstable as bagging.
- Boosting seems to be susceptible to noise.
  - When the number of outliners is very large, the emphasis placed on the hard examples can hurt the performance.

C4.5 and Boosting



Boosting over Reuters



Source: A Short Introduction to Boosting, (Freund&Schapire,99)
http://www.site.uottawa.ca/~stan/csi5387/boost-tut-ppr.pdf

Chapter 5: Partially-Supervised Learning

Learning from a small labeled set and a large unlabeled set

LU learning

Unlabeled Data

- One of the bottlenecks of classification is the labeling of a large set of examples (data records or text documents).
  - Often done manually
  - Time consuming
- Can we label only a small number of examples and make use of a large number of unlabeled examples to learn?
- Possible in many cases.

Why unlabeled data are useful?

- Unlabeled data are usually plentiful, labeled data are expensive.
- Unlabeled data provide information about the joint probability distribution over words and collocations (in texts).

- We will use text classification to study this problem.

Labeled Data

Unlabeled Data

Documents containing “homework” tend to belong to the positive class

DocNo: k ClassLabel: Positive
……
……homework….
…

DocNo: m ClassLabel: Positive
……
……homework….
…

DocNo: n ClassLabel: Positive
……
……homework….
…

DocNo: x (ClassLabel: Positive)
……
……homework….
…lecture….

DocNo: y (ClassLabel: Positive)
……lecture…..
……homework….
…

DocNo: z ClassLabel: Positive
……
……homework….
……lecture….

How to use unlabeled data

- One way is to use the EM algorithm
  - EM: Expectation Maximization
- The EM algorithm is a popular iterative algorithm for maximum likelihood estimation in problems with missing data.
- The EM algorithm consists of two steps,
  - *Expectation* step, i.e., filling in the missing data
  - *Maximization* step – calculate a new maximum *a posteriori* estimate for the parameters.

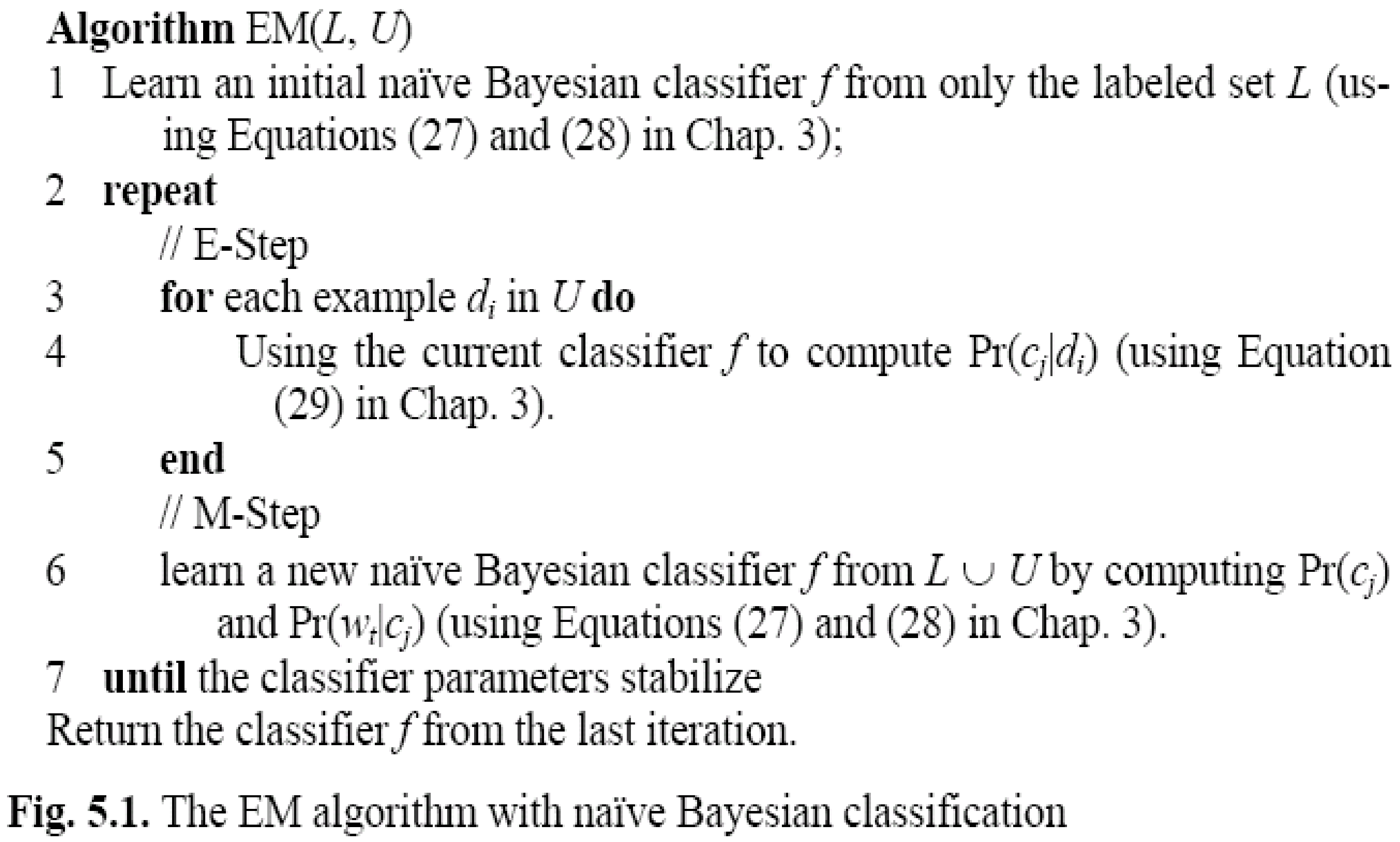
Incorporating unlabeled Data with EM (Nigam et al, 2000)

- Basic EM
- Augmented EM with weighted unlabeled data
- Augmented EM with multiple mixture components per class

Algorithm Outline

1. Train a classifier with only the labeled documents.
2. Use it to probabilistically classify the unlabeled documents.
3. Use ALL the documents to train a new classifier.
4. Iterate steps 2 and 3 to convergence.

Basic Algorithm



```
Algorithm EM(L, U)
1   Learn an initial naïve Bayesian classifier f from only the labeled set L (us-
        ing Equations (27) and (28) in Chap. 3);
2   repeat
        // E-Step
3       for each example $d_i$ in U do
4           Using the current classifier f to compute $Pr(c_j|d_i)$ (using Equation
              (29) in Chap. 3).
5       end
        // M-Step
6       learn a new naïve Bayesian classifier f from $L \cup U$ by computing $Pr(c_j)$
            and $Pr(w_t|c_j)$ (using Equations (27) and (28) in Chap. 3).
7   until the classifier parameters stabilize
Return the classifier f from the last iteration.
```

Fig. 5.1. The EM algorithm with naïve Bayesian classification

Basic EM: E Step & M Step

E Step:

$$\Pr(c_j \mid d_i; \hat{\Theta}) = \frac{\Pr(c_j \mid \hat{\Theta}) \Pr(d_i \mid c_j; \hat{\Theta})}{\Pr(d_i \mid \hat{\Theta})} \tag{29}$$

$$= \frac{\Pr(c_j \mid \hat{\Theta}) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_j; \hat{\Theta})}{\sum_{r=1}^{|C|} \Pr(c_r \mid \hat{\Theta}) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_r; \hat{\Theta})},$$

M Step:

$$\Pr(w_t \mid c_j; \hat{\Theta}) = \frac{\lambda + \sum_{i=1}^{|D|} N_{ti} \Pr(c_j \mid d_i)}{\lambda |V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N_{si} \Pr(c_j \mid d_i)}. \tag{27}$$

$$\Pr(c_j \mid \hat{\Theta}) = \frac{\sum_{i=1}^{|D|} \Pr(c_j \mid d_i)}{|D|}. \tag{28}$$

The problem

- It has been shown that the EM algorithm in Fig. 5.1 works well if the
  - The two mixture model assumptions for a particular data set are true.
- The two mixture model assumptions, however, can cause major problems when they do not hold. In many real-life situations, they may be violated.
- It is often the case that a class (or topic) contains a number of sub-classes (or sub-topics).
  - For example, the class Sports may contain documents about different sub-classes of sports, Baseball, Basketball, Tennis, and Softball.
- Some methods to deal with the problem.

Weighting the influence of unlabeled examples by factor μ

New M step:

$$\Pr(w_t \mid c_j) = \frac{\lambda + \sum_{i=1}^{|D|} \Lambda(i) N_{ti} \Pr(c_j \mid d_i)}{\lambda |V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} \Lambda(i) N_{ti} \Pr(c_j \mid d_i)}, \qquad (1)$$

where

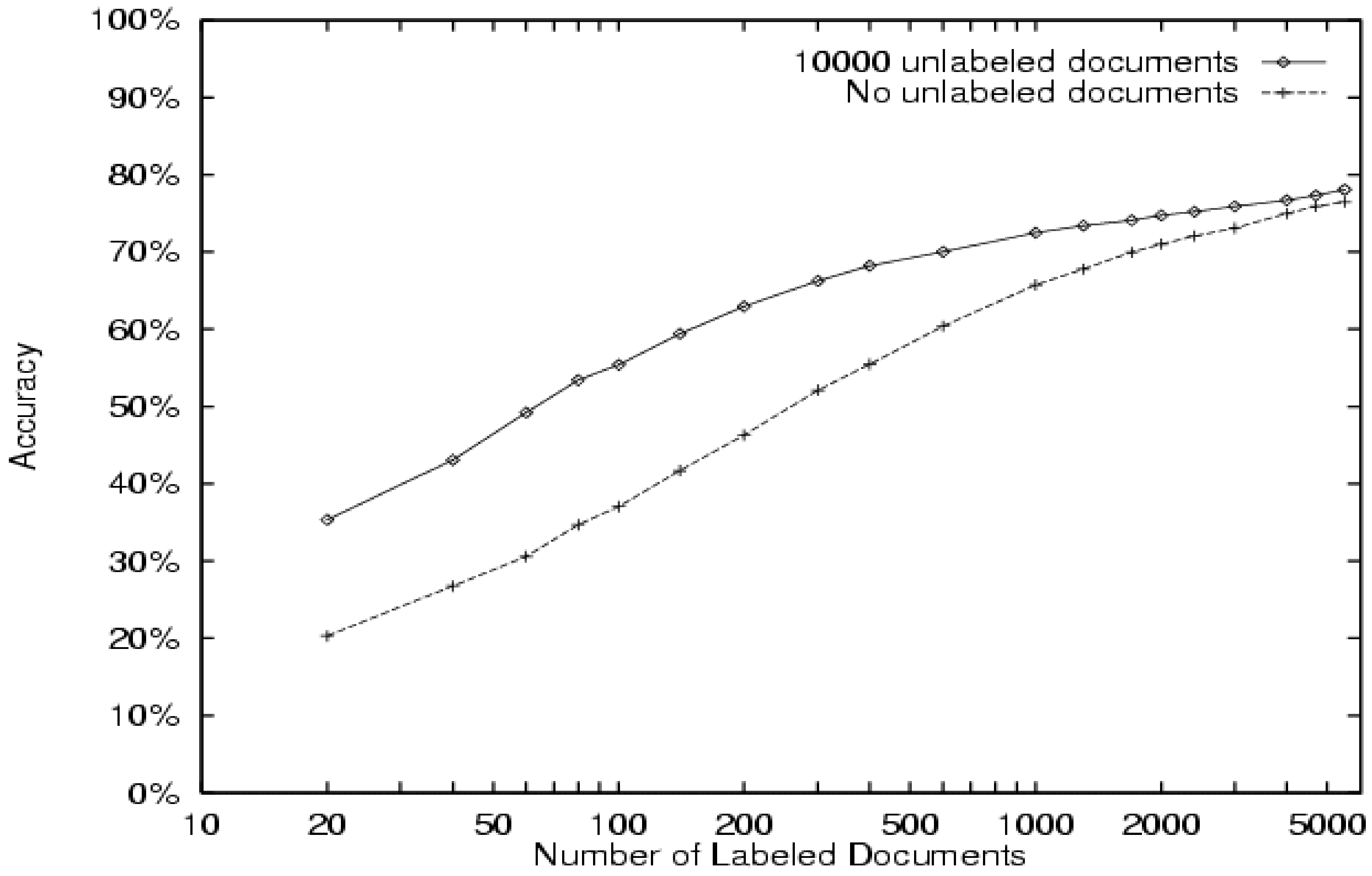
$$\Lambda(i) = \begin{cases} \mu & \text{if } d_i \in U \\ 1 & \text{if } d_i \in L. \end{cases} \qquad (2)$$

The prior probability also needs to be weighted.

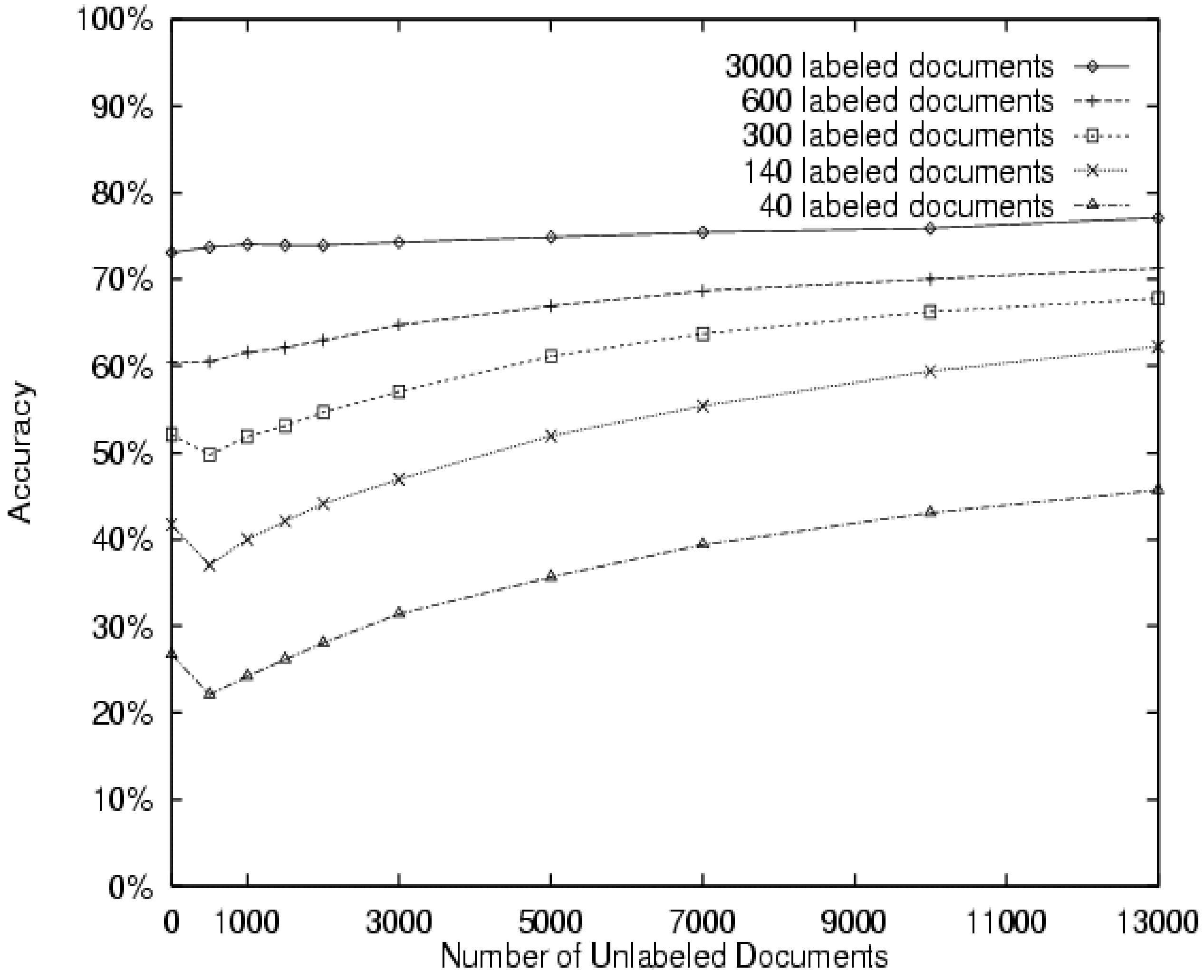
Experimental Evaluation

- Newsgroup postings
  - 20 newsgroups, 1000/group
- Web page classification
  - student, faculty, course, project
  - 4199 web pages
- Reuters newswire articles
  - 12,902 articles
  - 10 main topic categories

20 Newsgroups



20 Newsgroups



Another approach: Co-training

- Again, learning with a small labeled set and a large unlabeled set.
- The attributes describing each example or instance can be partitioned into two subsets. Each of them is sufficient for learning the target function.
  - E.g., hyperlinks and page contents in Web page classification.
- Two classifiers can be learned from the same data.

Co-training Algorithm
[Blum and Mitchell, 1998]

```
Given: labeled data L,
       unlabeled data U
Loop:
    Train h1 (e.g., hyperlink classifier) using L
    Train h2 (e.g., page classifier) using L
    Allow h1 to label p positive, n negative examples from U
    Allow h2 to label p positive, n negative examples from U
    Add these most confident self-labeled examples to L
```

Co-training: Experimental Results

- begin with 12 labeled web pages (academic course)
- provide 1,000 additional unlabeled web pages
- average error: learning from labeled data 11.1%;
- average error: co-training 5.0%

	Page-base classifier	Link-based classifier	Combined classifier
Supervised training	12.9	12.4	11.1
Co-training	6.2	11.6	5.0

Co-training: Experimental Results

- begin with 12 labeled web pages (academic course)
- provide 1,000 additional unlabeled web pages
- average error: learning from labeled data 11.1%;
- average error: co-training 5.0%

	Page-base classifier	Link-based classifier	Combined classifier
Supervised training	12.9	12.4	11.1
Co-training	6.2	11.6	5.0

When the generative model is not suitable

- Multiple Mixture Components per Class (M-EM). E.g., a class --- a number of sub-topics or clusters.
- Results of an example using 20 newsgroup data
  - 40 labeled; 2360 unlabeled; 1600 test
  - Accuracy
    - NB 68%
    - EM 59.6%
- Solutions
  - M-EM (Nigam et al, 2000): Cross-validation on the training data to determine the number of components.
  - Partitioned-EM (Cong, et al, 2004): using hierarchical clustering. It does significantly better than M-EM.

Summary

- Using unlabeled data can improve the accuracy of classifier when the data fits the generative model.
- Partitioned EM and the EM classifier based on multiple mixture components model (M-EM) are more suitable for real data when multiple mixture components are in one class.
- Co-training is another effective technique when redundantly sufficient features are available.

Further Topics

- Learning from Positive and Unlabeled Example (PU).

- Graph-based methods for Semi-supervised learning
  - Labeled and unlabeled examples are nodes in a graph
  - **mincut**: See the labeling of Us as a graph partition process (polynomial time)
  - **Spectral Graph transducer**: map the graph partition into a minimization problem and apply eigenvector analysis to find the best solutions. Parameters: balancing factors between P and U instances

- ICML '07 Tutorial (by Jerry Zhu) at:
  `http://pages.cs.wisc.edu/~jerryzhu/icml07tutorial.html`