Text Classification from Labelled and Unlabelled Documents using EM

- **Paper by**
  Kamal Nigam,
  Andrew Kachites McCallum,
  Sebastian Thrun,
  Tom Mitchel

- **Presentation by**

- Ruoxun Fu
- Carlos Fornieles Montoya

Introduction

- Typically we want to classify documents given a large enough (labelled) training data set.

- Given enough labelled examples we can build an accurate classifier.

- Finding examples is not a problem.

Introduction 2

- The problem is to have someone label sufficient examples for us to train the classifier.

- Thousands of labelled documents needed in order to build an accurate classifier. Infeasible.

- It is possible to use unlabelled data to train the classifier and improve its accuracy.

Algorithm outline

- Train a classifier with the available labelled documents.

- Use this classifier to assign a probabilistically-weighted class label to the unlabelled documents.

- Finally train a new classifier using all the documents.

Algorithm Outline 2

- The algorithm uses a combination of EM and naive Bayes.
- Basic algorithm makes 2 assumptions:
  - Data has been generated with a mixture model.
  - There is a one to one correspondence between mixture components and classes.
- These assumptions rarely hold.

Algorithm Outline 3

- In order to cope with this, the algorithm is extended with:
  - Weighting factor that adjusts the strength of the unlabelled data's contribution to parameter estimation in EM.
  - Reduce the bias of naive Bayes by modelling each class with multiple mixture components.

- We will talk about this in detail later.

Unlabelled Data

- Unlabelled data alone is useless.
- But combined with labelled data it can be very useful:
  - In the text classification case, it provides information about the joint probability distribution over words.
  - For example if the word “homework” gives us important evidence of some class, we may find that the word “lecture” also appears in those documents with the unlabelled data.

Unlabelled Data 2

- Another example: It is known that a Gaussian mixture model with 2 components can be recovered with just unlabelled data.
- Problem is that without labelled data we cannot assign a class to each mixture component.

Notation

- Let d_i denote the ith document.
- Let θ denote the set of parameters of the probability distribution that generates the documents.
- Probability distribution consists of a mixture of components $c_j \in C = \{c_1, \ldots, c_{|C|}\}$
- Let y_i denote the label for the ith document.
- Let V denote our vocabulary of words.

First Step: Naive Bayes

- First step consists of training a Naive Bayes classifier with the labelled data.
- Probability of a document being generated by the probability distribution:

$$P(d_i|c_j,\theta)=P(|d_i|)\prod_{k=1}^{|d_i|}P(w_{d_{i,k}}|c_j,\theta) \qquad (4)$$

- This further assumes the following things:
  - Document length is independent of the document class.
  - Words are independent of each other.
  - And the order in which they appear is unimportant.

First Step: Naive Bayes 2

- Now it makes sense to model each mixture component with a multinomial distribution, so we can define the set of parameters

  $$\theta = \{ \theta_{w_t|c_j} : w_t \in V, c_j \in C, \theta_{c_j} : c_j \in C \}$$

  - $\theta_{w_t|c_j} = P(w_t|c_j, \theta)$ are the parameters of the multinomial distribution for component c_j.
  - c_j are the different mixture components
  - And each θ_{c_j} is the probability of component c_j being chosen.

Training the Naive Bayes classifier

- Let D be the set of labelled documents.
- The estimates for the parameters of the distribution are calculated using maximum a posteriori parameter estimation:

$$\hat{\theta}_{w_t|c_j} \equiv P(w_t|c_j, \hat{\theta}) = \frac{1 + \sum_{i=1}^{|D|} N(w_t, d_i) P(y_i = c_j|d_i)}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N(w_s, d_i) P(y_i = c_j|d_i)} \tag{5}$$

$N(w_t, d_i)$ is the number of times that word w_t appears in d_i

$$\hat{\theta}_{c_j} \equiv P(c_j|\hat{\theta}) = \frac{1 + \sum_{i=1}^{|D|} P(y_i = c_j|d_i)}{|C| + |D|} \tag{6}$$

Using the classifier

- By applying Bayes rule and substituting we get:

$$P(y_i=c_j|d_i,\hat{\theta})=\frac{P(c_j|\hat{\theta})P(d_i|c_j,\hat{\theta})}{P(d_i|\hat{\theta})}=$$

$$=\frac{P(c_j|\hat{\theta})\prod_{k=1}^{|d_i|}P(w_{d_{i,k}}|c_j,\hat{\theta})}{\sum_{r=1}^{|C|}P(c_r|\hat{\theta})\prod_{k=1}^{|d_i|}P(w_{d_{i,k}}|c_r,\hat{\theta})} \qquad (7)$$

- Even though the assumptions we made do not hold in the real-world, Naive Bayes does a good job. The independence assumption causes the probability estimates to be extreme (almost 0 or 1) but this is not incompatible with having good performance as a classifier.

Second Step: EM

- For the second step, we assigned probability weighted class labels to the unlabelled data.
- Let $D = D^l \cup D^u$ where D^l is the set of labelled documents and D^u is the set of unlabelled documents. The likelihood for all the data is:

$$P(D|\theta) = \prod_{d_i \in D^u} \sum_{j=1}^{|C|} P(c_j|\theta) P(d_i|c_j, \theta) \times \\ \times \prod_{d_i \in D^l} P(y_i = c_j|\theta) P(d_i|y_i = c_j, \theta) \tag{8}$$

Second Step: EM 2

- But we will work with $\log P(\theta|D)$, in stead of working with $\log P(D|\theta)$ directly. Note that $P(\theta|D) \propto P(\theta) P(D|\theta)$.
- Now we take logs:

$$\log(P(\theta)P(D|\theta)) = \log(P(\theta)) + \sum_{d_i \in D^u} \log \sum_{j=1}^{|C|} P(c_j|\theta) P(d_i|c_j, \theta) + \\ + \sum_{d_i \in D^l} P(y_i = c_j|\theta) P(d_i|y_i = c_j, \theta) \qquad (9)$$

- This expression is very hard to maximize. We need a workaround.

Second Step: EM Workaround

- If we knew the class for each document, e.g. we have a matrix of elements $z_{i,j}$ such that $z_{i,j}$ is 1 if and only if the class for the ith document is class c_j and 0 otherwise, then we can write the previous expression as:

$$l_c(\theta|D,\boldsymbol{z})=\log(P(\theta))+\sum_{d_i\in D}\sum_{j=1}^{|C|} z_{i,j}\log(P(c_j|\theta)P(d_i|c_j,\theta)) \tag{10}$$

Second Step: EM Workaround 2

- The previous equation bounds (9) from below. And unlike (9), this equation can be easily maximized.
- Now we can perform EM with this equation:
  - While the classifier's parameters improve:
    - E-step: $\hat{\boldsymbol{z}}^{k+1} = E[\boldsymbol{z} | D, \hat{\theta}^k]$, i.e.
      Use current classifier $\hat{\theta}$ to estimate the probability that each mixture component generated each unlabelled document, $P(c_j | d_i, \hat{\theta})$
    - M-step: $\hat{\theta}^{k+1} = arg\,max_{\theta} P(\theta | D, \hat{z}^{k+1})$, i.e.
      Re-estimate the parameters $\hat{\theta}$ by using the same procedure that we used earlier to train the classifier, but considering the estimations for the unlabelled data.

Discussion

- Apparently, in some cases using unlabelled data greatly improves classification accuracy.
- But there are examples where using unlabelled data actually degrades accuracy.
- In the next part we will see how to get around this problem and experimental results.

Augmented EM

- Assumption of basic EM
- Weighting the unlabelled data
- Multiple mixture components per class

Assumptions of basic EM

- The data is produced by a mixture model.
- There is a one-to-one correspondence between mixture components and classes.
- These assumptions are usually violated by real-world textual data.
- The benefits of unlabelled data is less clear when the assumptions don’t hold.

Assumptions of basic EM

- 2 extensions of basic EM for violated assumptions:
  - Adjusting the influence of unlabeled the data-EM-λ Algorithm.
  - Relaxing the “one-to-one” constraint-Multiple mixture components per class, many-to-one mapping.

Extension 1: Weighting the unlabeled data

- In common scenario, the unlabelled data are far more than labelled data.
- EM usually performs a role of unsupervised natural clustering.
- The unlabelled data set mainly determines the classifier’s parameters.
- The labelled data only initialize classifier’s parameters and identify each component with a class label.

Extension 1: Weighting the unlabeled data

- When the mixture model assumptions are not true, the natural clustering of the unlabelled data may produce mixture components that are not in correspondence with class labels.
- This effect is particularly apparent when we have enough labelled data to obtain a good parameter estimates.

Extension 1: Weighting the unlabeled data

- The idea is modulating the influence of unlabeled data when EM performs unsupervised clustering.
- We introduce a λ parameter to decrease the contribution of unlabeled data in log likelihood equation:

$$
\begin{aligned}
l_c(\theta|\mathcal{D};\mathbf{z}) \;=\; & \log(\mathrm{P}(\theta)) + \sum_{d_i\in\mathcal{D}^l}\sum_{j=1}^{|\mathcal{C}|} z_{ij}\log\left(\mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta)\right) \\
& + \lambda\left(\sum_{d_i\in\mathcal{D}^u}\sum_{j=1}^{|\mathcal{C}|} z_{ij}\log\left(\mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta)\right)\right).
\end{aligned}
$$

Extension 1: Weighting the unlabeled data

- When λ=0, the unlabeled data have litter influence on parameter estimates.
- When λ=1, the unlabeled data will be weighted the same as labeled ones.
- The E-Step is performed exactly the same as before.
- The M-Step is different, first define Λ(i):

$$\Lambda(i) = \begin{cases} \lambda & \text{if } d_i \in \mathcal{D}^u \\ 1 & \text{if } d_i \in \mathcal{D}^l. \end{cases}$$

Extension 1: Weighting the unlabeled data

- Then the equations in M-Step are changed into:

$$\hat{\theta}_{w_t|c_j} \equiv \mathrm{P}(w_t|c_j;\hat{\theta}) = \frac{1+\sum_{i=1}^{|\mathcal{D}|}\Lambda(i)N(w_t,d_i)\mathrm{P}(y_i=c_j|d_i)}{|V|+\sum_{s=1}^{|V|}\sum_{i=1}^{|\mathcal{D}|}\Lambda(i)N(w_s,d_i)\mathrm{P}(y_i=c_j|d_i)}.$$

$$\hat{\theta}_{c_j} \equiv \mathrm{P}(c_j|\hat{\theta}) = \frac{1+\sum_{i=1}^{|\mathcal{D}|}\Lambda(i)\mathrm{P}(y_i=c_j|d_i)}{|\mathcal{C}|+|\mathcal{D}^l|+\lambda|\mathcal{D}^u|}.$$

Extension 1: Weighting the unlabeled data

- In experiments we select λ maximizes the leave-one-out cross-validation classification accuracy of the labeled data.

- Experimental results show that setting λ to some value between 0 and 1 can result in higher classification accuracy than either λ=0 or 1, even when unlabeled data's natural clustering would result in poor classification

Extension 2: Multiple Mixture components per class

- Another idea is to relax the “one-to-one” assumption by replacing it with a “many-to-one” assumption.
- A class can be represented by multiple components.
- For textual data, this means that a class can be comprised of several different sub-topics, each with a different word distribution.

t_a

Extension 2: Multiple Mixture components per class

- Using multiple mixture components per class can capture some dependencies between words.
- This method introduces “missing values” for both labeled data and unlabeled data.
- We will now write t_a for the *a*th class, and c_j will continue to denote the *j*th component.

t_a

Extension 2: Multiple Mixture components per class

- We write $\mathrm{P}(t_a|c_j;\hat{\theta}) \in \{0,1\}$ for the pre-determined many-to-one mapping between mixture components and classes.
- The M-Step is the same as basic EM.
- In the E-Step, unlabelled data are treated as before.
- In the E-Step, $\mathrm{P}(c_j|d_i;\hat{\theta})$ are normalized to sum to one for $\mathrm{P}(y_i = t_a|c_j;\hat{\theta})$ is 1, otherwise 0.

t_a

Extension 2: Multiple Mixture components per class

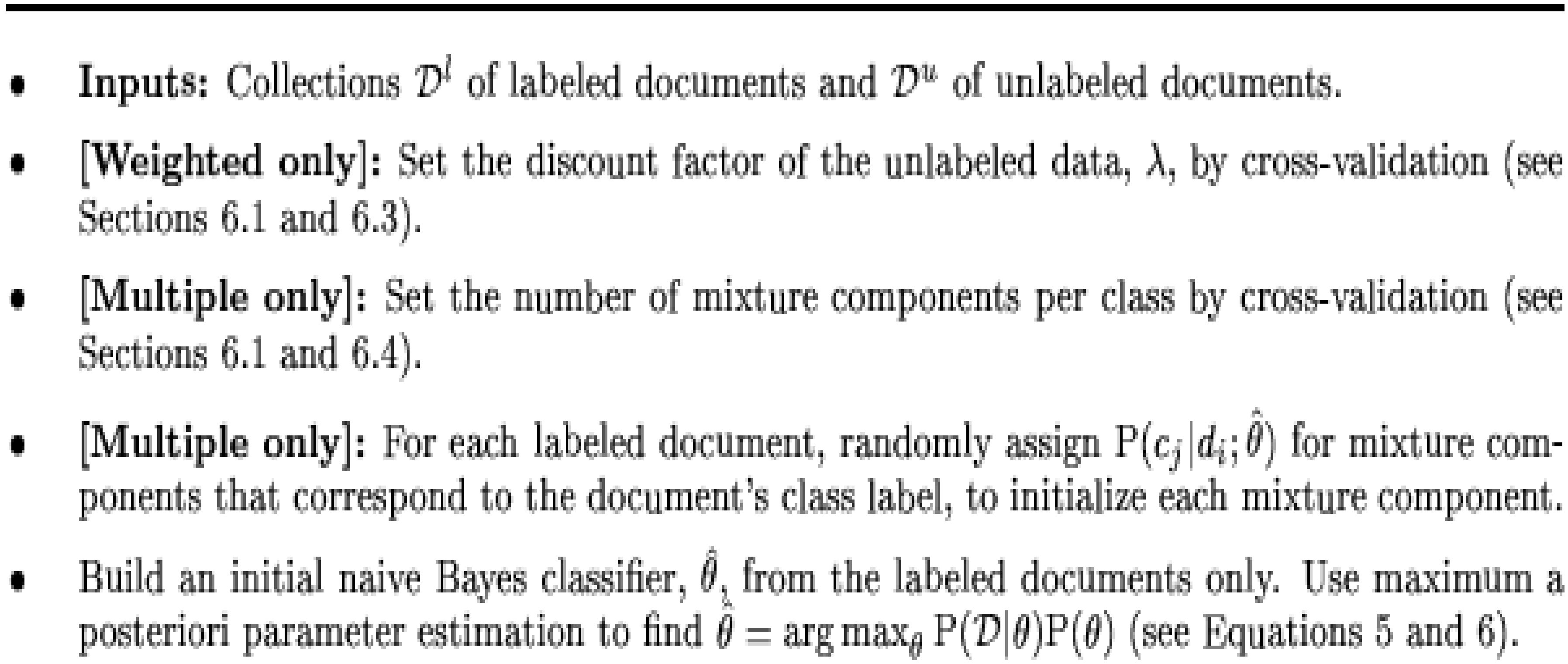
- Components are initialized by performing a randomized E-step in which $\mathrm{P}(c_j|d_i;\hat{\theta})$ is sampled from a uniform distribution.
- Classification becomes a matter of probabilistically “classifying” documents into the mixture components, and then summing into class probabilities:

$$\mathrm{P}(t_a|d_i;\hat{\theta}) = \sum_{c_j} \mathrm{P}(t_a|c_j;\hat{\theta}) \frac{\mathrm{P}(c_j|\hat{\theta}) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{i,k}}|c_j;\hat{\theta})}{\sum_{r=1}^{|C|} \mathrm{P}(c_r|\hat{\theta}) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{i,k}}|c_r;\hat{\theta})}.$$

t_a

Extension 2: Multiple Mixture components per class

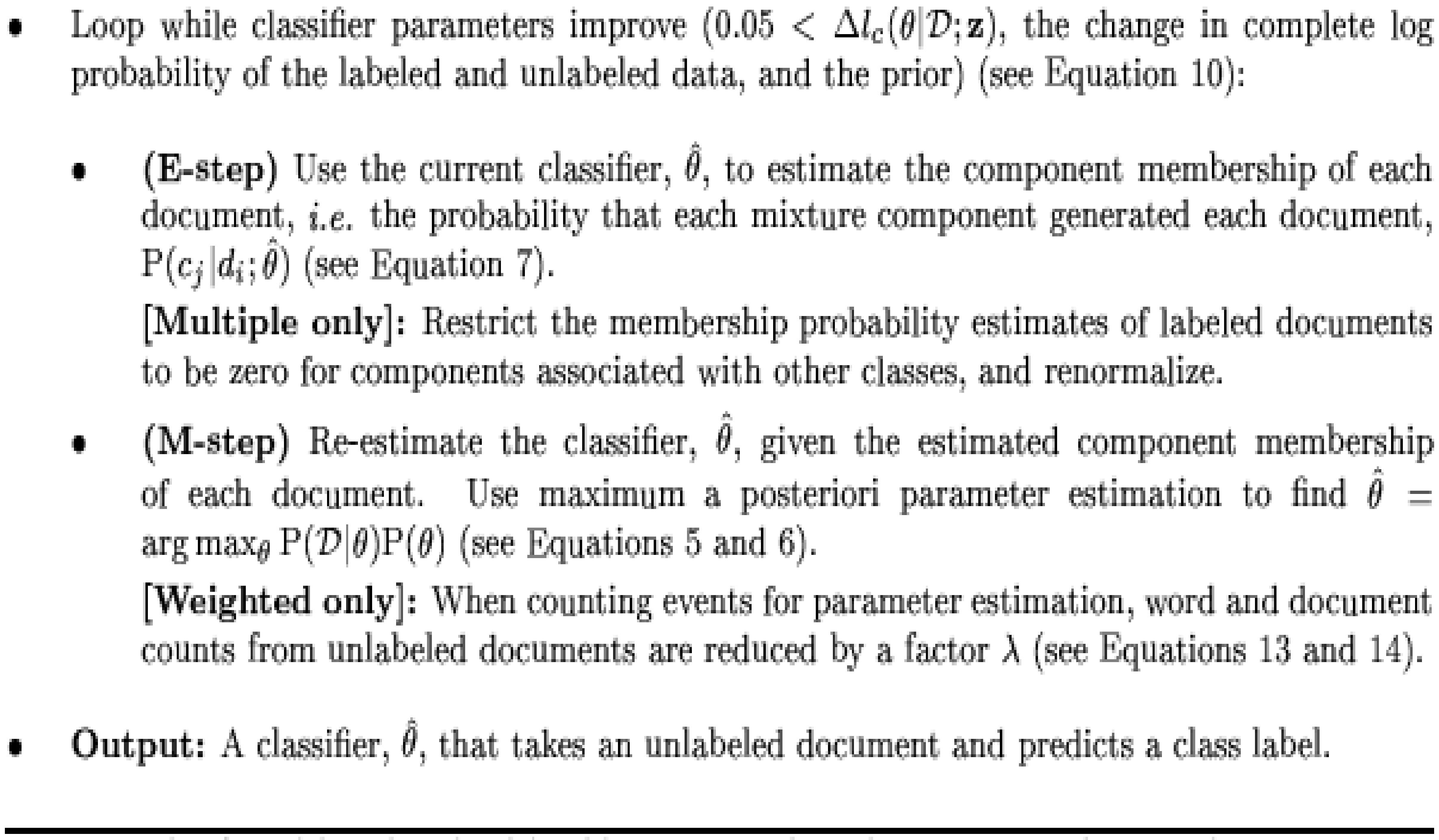
- Algorithm:



- **Inputs:** Collections $\mathcal{D}^l$ of labeled documents and $\mathcal{D}^u$ of unlabeled documents.
- **[Weighted only]:** Set the discount factor of the unlabeled data, λ, by cross-validation (see Sections 6.1 and 6.3).
- **[Multiple only]:** Set the number of mixture components per class by cross-validation (see Sections 6.1 and 6.4).
- **[Multiple only]:** For each labeled document, randomly assign $\mathrm{P}(c_j|d_i;\hat{\theta})$ for mixture components that correspond to the document's class label, to initialize each mixture component.
- Build an initial naive Bayes classifier, $\hat{\theta}$, from the labeled documents only. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg\max_\theta \mathrm{P}(\mathcal{D}|\theta)\mathrm{P}(\theta)$ (see Equations 5 and 6).

t_a

Extension 2: Multiple Mixture components per class



- Loop while classifier parameters improve ($0.05 < \Delta l_c(\theta|\mathcal{D};\mathbf{z})$, the change in complete log probability of the labeled and unlabeled data, and the prior) (see Equation 10):
  - **(E-step)** Use the current classifier, $\hat{\theta}$, to estimate the component membership of each document, *i.e.* the probability that each mixture component generated each document, $\mathrm{P}(c_j|d_i;\hat{\theta})$ (see Equation 7).

    [Multiple only]: Restrict the membership probability estimates of labeled documents to be zero for components associated with other classes, and renormalize.
  - **(M-step)** Re-estimate the classifier, $\hat{\theta}$, given the estimated component membership of each document. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg\max_\theta \mathrm{P}(\mathcal{D}|\theta)\mathrm{P}(\theta)$ (see Equations 5 and 6).

    [Weighted only]: When counting events for parameter estimation, word and document counts from unlabeled documents are reduced by a factor λ (see Equations 13 and 14).
- **Output:** A classifier, $\hat{\theta}$, that takes an unlabeled document and predicts a class label.

Experimental Test

- Empirical evidence is provided by comparing traditional naive bayes and using EM with labeled and unlabeled data.
- Generally, the improvements in accuracy due to unlabeled data are dramatic, especially when the number of labeled data is low.
- The augmented EM can increase performance even when basic EM performs poor.

Data Sets

- 20 Newsgroup:
  - 20017 articles divided evenly among 20 different UseNet discussion groups.
  - The task is to classify an article into the one newsgroup to which it was posted.
  - Many categories fall into confusable clusters.
  - Stop words are removed.
  - Word counts are normalized and scaled such that each document has constant length.

Data Sets

- WebKB:
  - 8145 Web pages gathered from university computer science departments.
  - Choosing 4199 pages covering categories: student, faculty, course and project.
  - The task is to classify a web page into one of the four categories.
  - Stemming and stoplist are not used.
  - Vocabulary is limited to 300 most informative words.

Data Sets

- Reuters:
  - 12902 articles and 90 topic categories.
  - The task is to build a binary classifier for each of the ten most populous classes to identify the news topic.
  - Words inside <TEXT> tags are used.
  - Stoplist are used, but no stemming.
  - The vocabulary size varies for different category.
  - Metrics are Recall and Precision instead of Accuracy.

$$\begin{aligned}\text{Recall} &= \frac{\#\text{ of correct positive predictions}}{\#\text{ of positive examples}}\\ \text{Precision} &= \frac{\#\text{ of correct positive predictions}}{\#\text{ of positive predictions}}.\end{aligned}$$

Practical Computation

- A simplified leave-one-out cross-validation is performed in conjunction with EM.
- The computation complexity is not prohibitive.
- In experiments, EM usually converges after 10 iterations.
- The process of task is less than 1 minute for the WebKB, less than 15 minutes for 20 newsgroups,

EM with Unlabeled Data Increases Accuracy

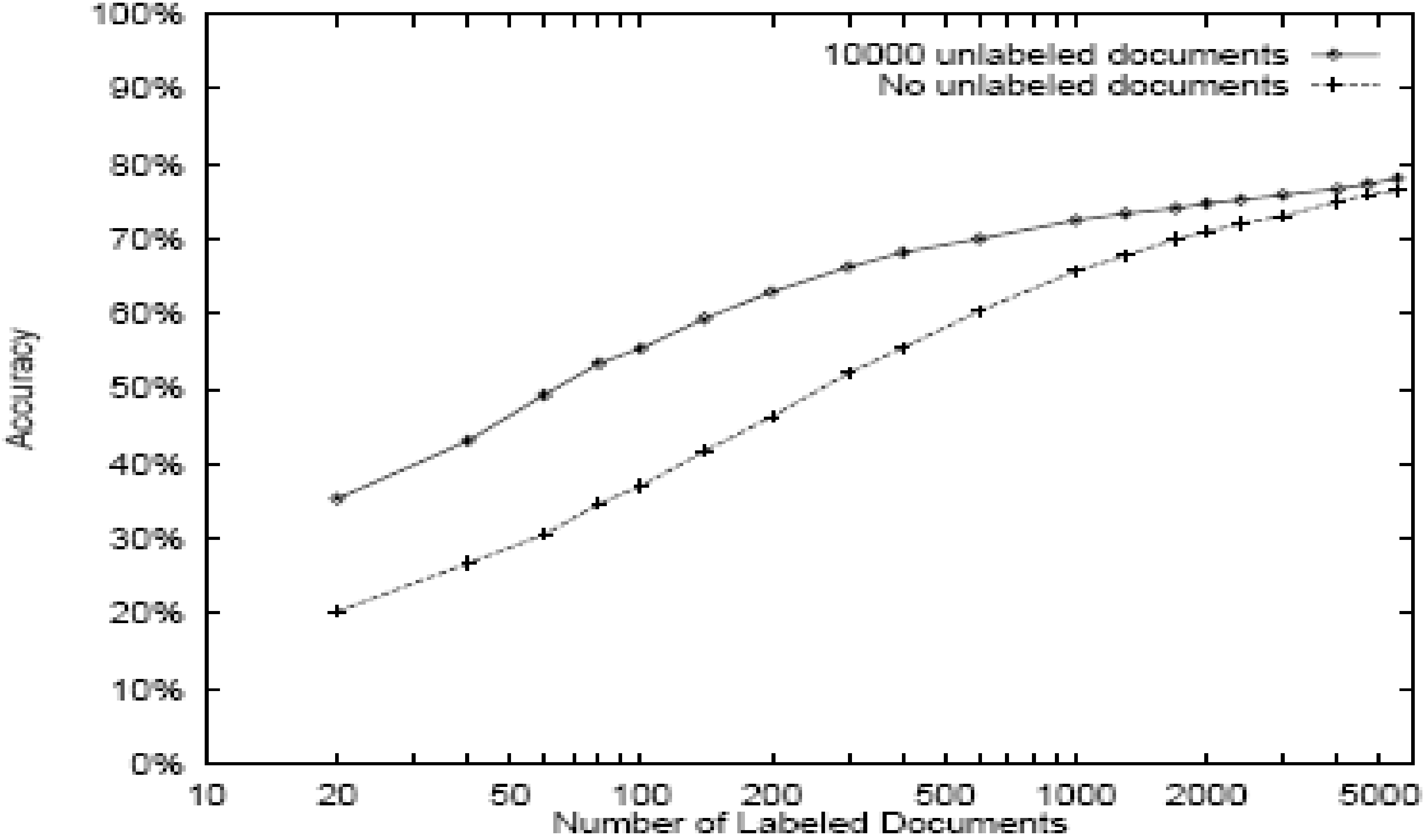


Figure 2. Classification accuracy on the 20 Newsgroups data set, both with and without 10,000 unlabeled documents. With small amounts of training data, using EM yields more accurate classifiers. With large amounts of labeled training data, accurate parameter estimates can be obtained without the use of unlabeled data, and the two methods begin to converge.

The influence of different labeled data

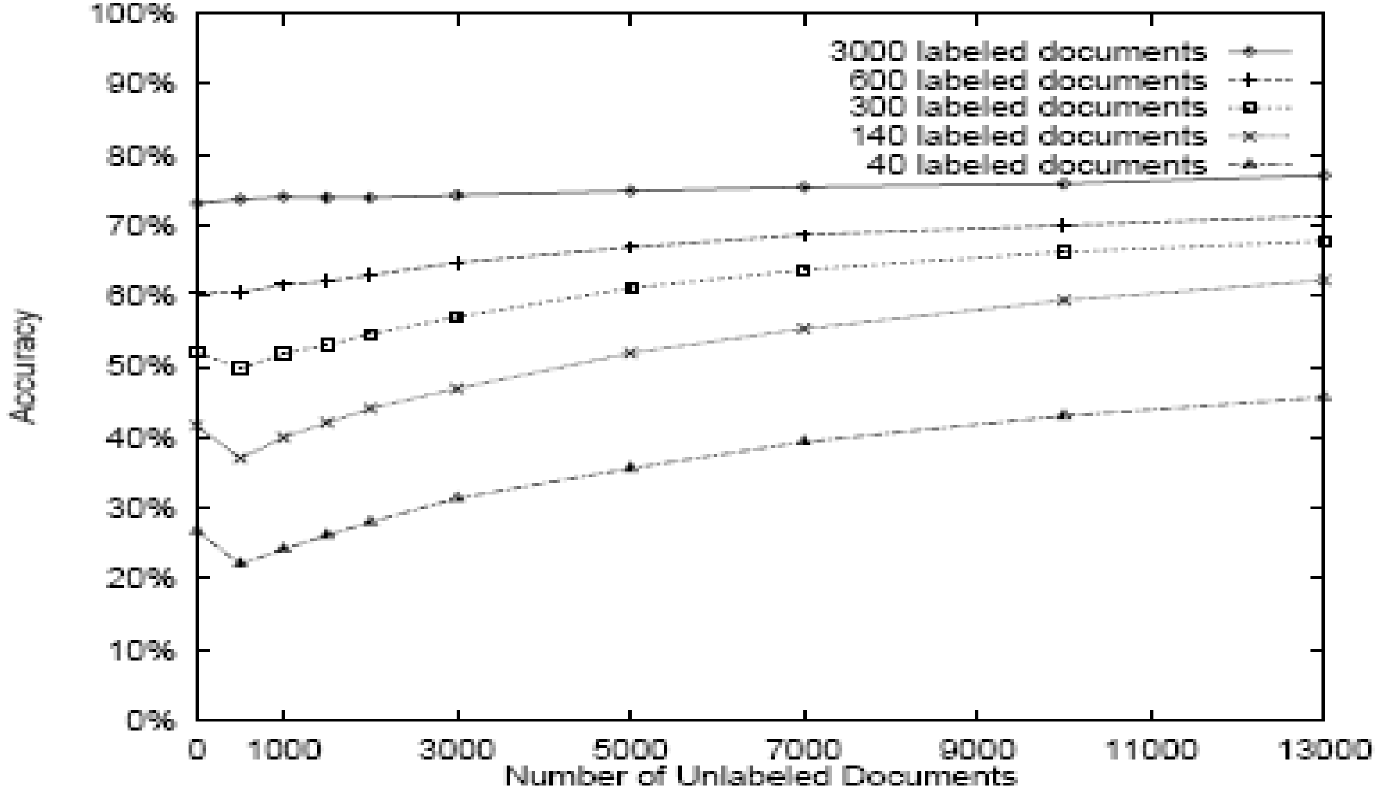


Figure 3. Classification accuracy while varying the number of unlabeled documents. The effect is shown on the 20 Newsgroups data set, with 5 different amounts of labeled documents, by varying the amount of unlabeled data on the horizontal axis. Having more unlabeled data helps. Note the dip in accuracy when a small amount of unlabeled data is added to a small amount of labeled data. We hypothesize that this is caused by extreme, almost 0 or 1, estimates of component membership, $\mathrm{P}(c_j|d_i,\hat{\theta})$, for the unlabeled documents (as caused by naive Bayes' word independence assumption).

Tracking the EM

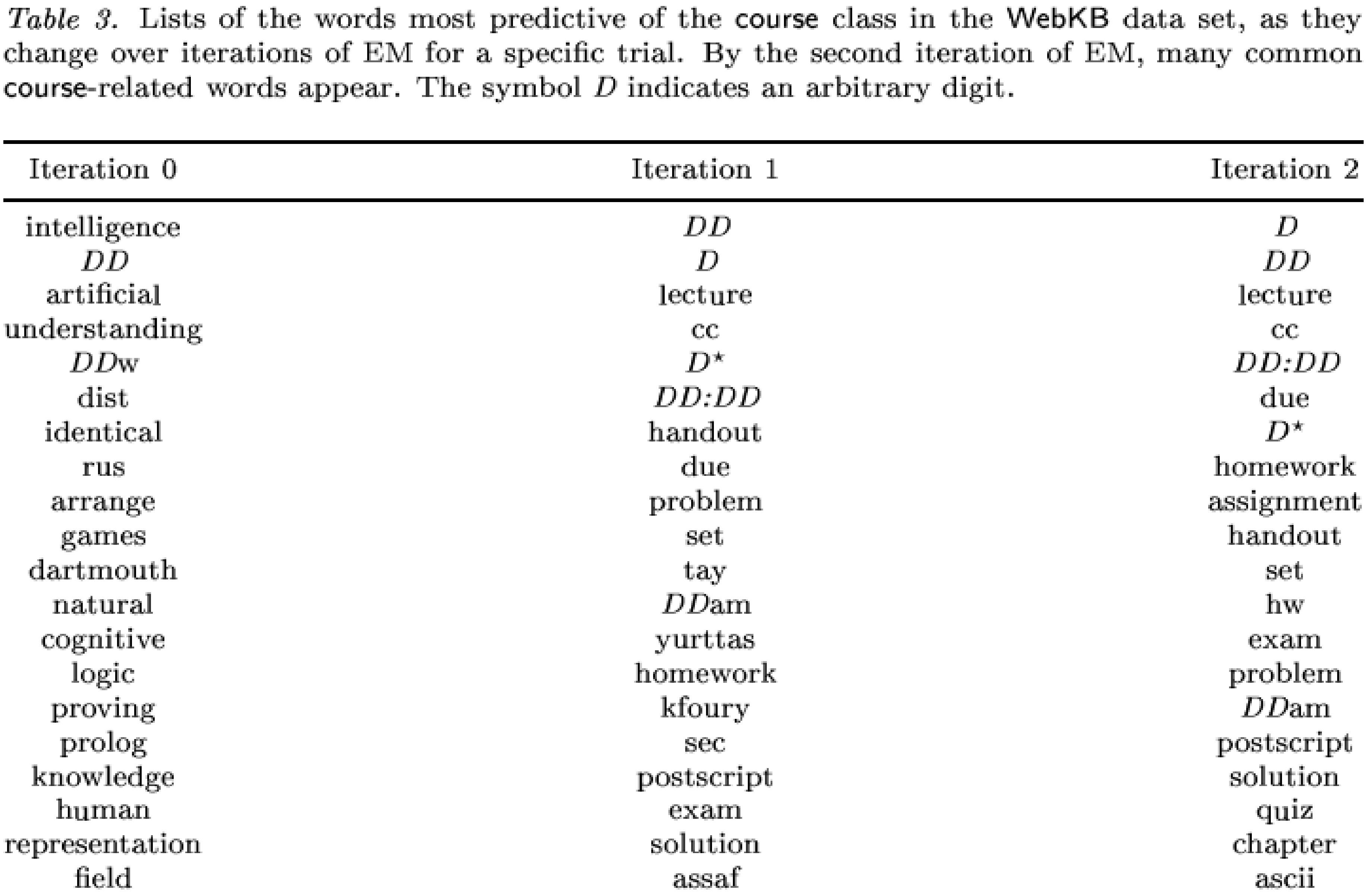


Table 3. Lists of the words most predictive of the course class in the WebKB data set, as they change over iterations of EM for a specific trial. By the second iteration of EM, many common course-related words appear. The symbol D indicates an arbitrary digit.

Iteration 0	Iteration 1	Iteration 2
intelligence	DD	D
DD	D	DD
artificial	lecture	lecture
understanding	cc	cc
DDw	$D^\star$	$DD{:}DD$
dist	$DD{:}DD$	due
identical	handout	$D^\star$
rus	due	homework
arrange	problem	assignment
games	set	handout
dartmouth	tay	set
natural	DDam	hw
cognitive	yurttas	exam
logic	homework	problem
proving	kfoury	DDam
prolog	sec	postscript
knowledge	postscript	solution
human	exam	quiz
representation	solution	chapter
field	assaf	ascii

EM may decrease performance

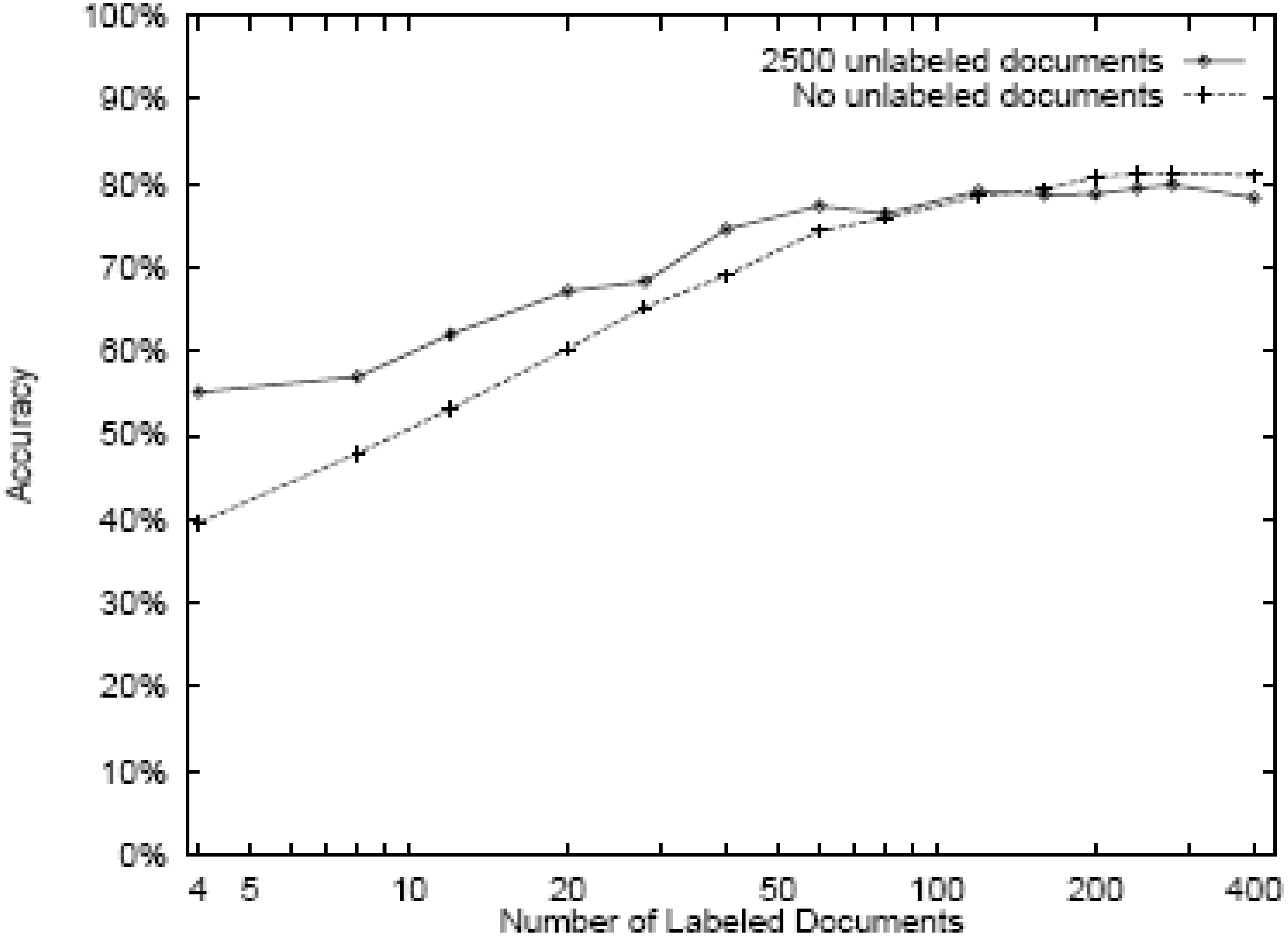


Figure 4. Classification accuracy on the WebKB data set, both with and without 2500 unlabeled documents. When there are small numbers of labeled documents, EM improves accuracy. When there are many labeled documents, however, EM degrades performance slightly—indicating a misfit between the data and the assumed generative model.

Weighting unlabeled data

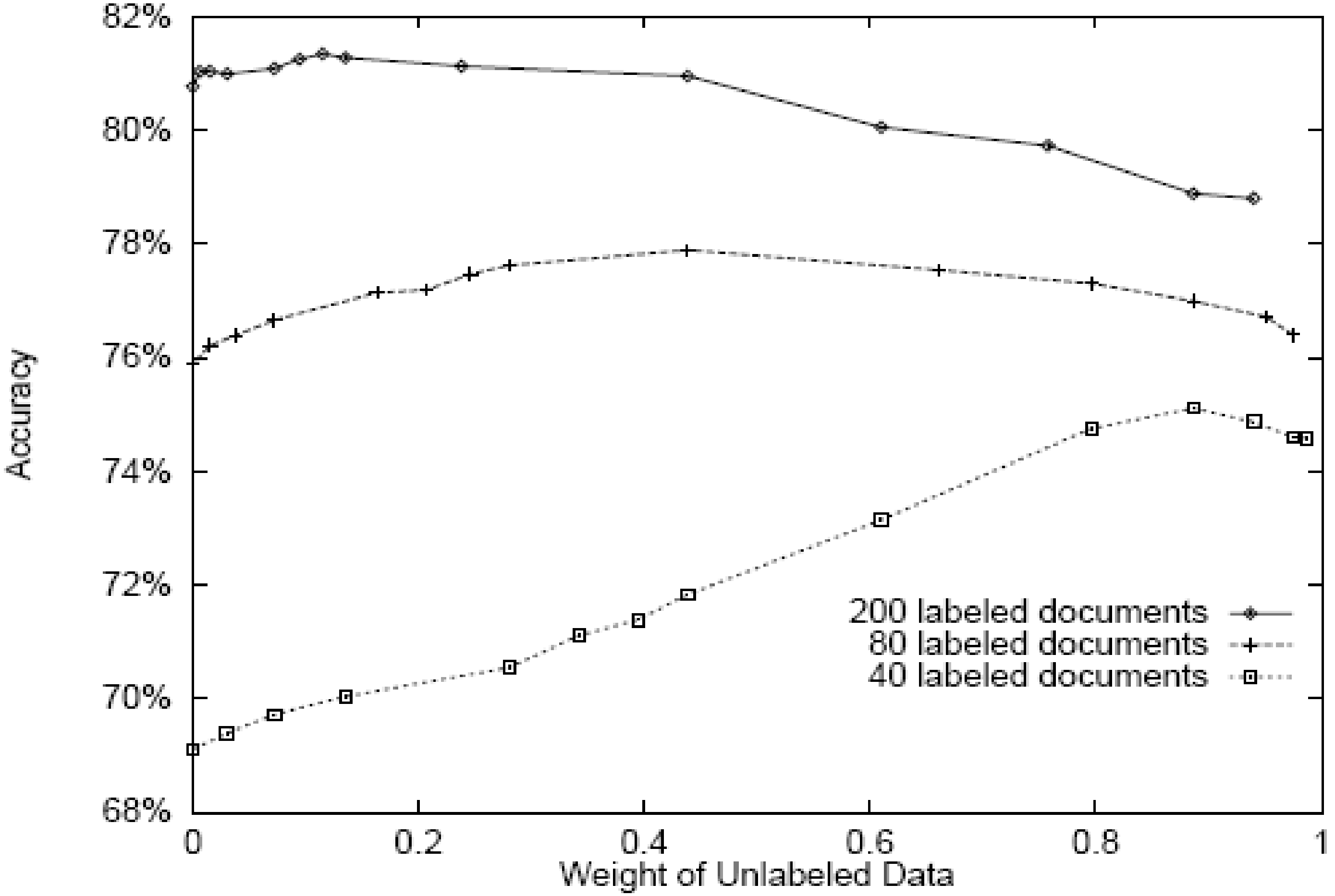


Figure 5. The effects of varying λ, the weighting factor on the unlabeled data in EM-λ. These three curves from the WebKB data set correspond to three different amounts of labeled data. When there is less labeled data, accuracy is highest when more weight is given to the unlabeled data. When the amount of labeled data is large, accurate parameter estimates are attainable from the labeled data alone, and the unlabeled data should receive less weight. With moderate amounts of labeled data, accuracy is better in the middle than at either extreme. Note the magnified vertical scale.

Different EM comparing

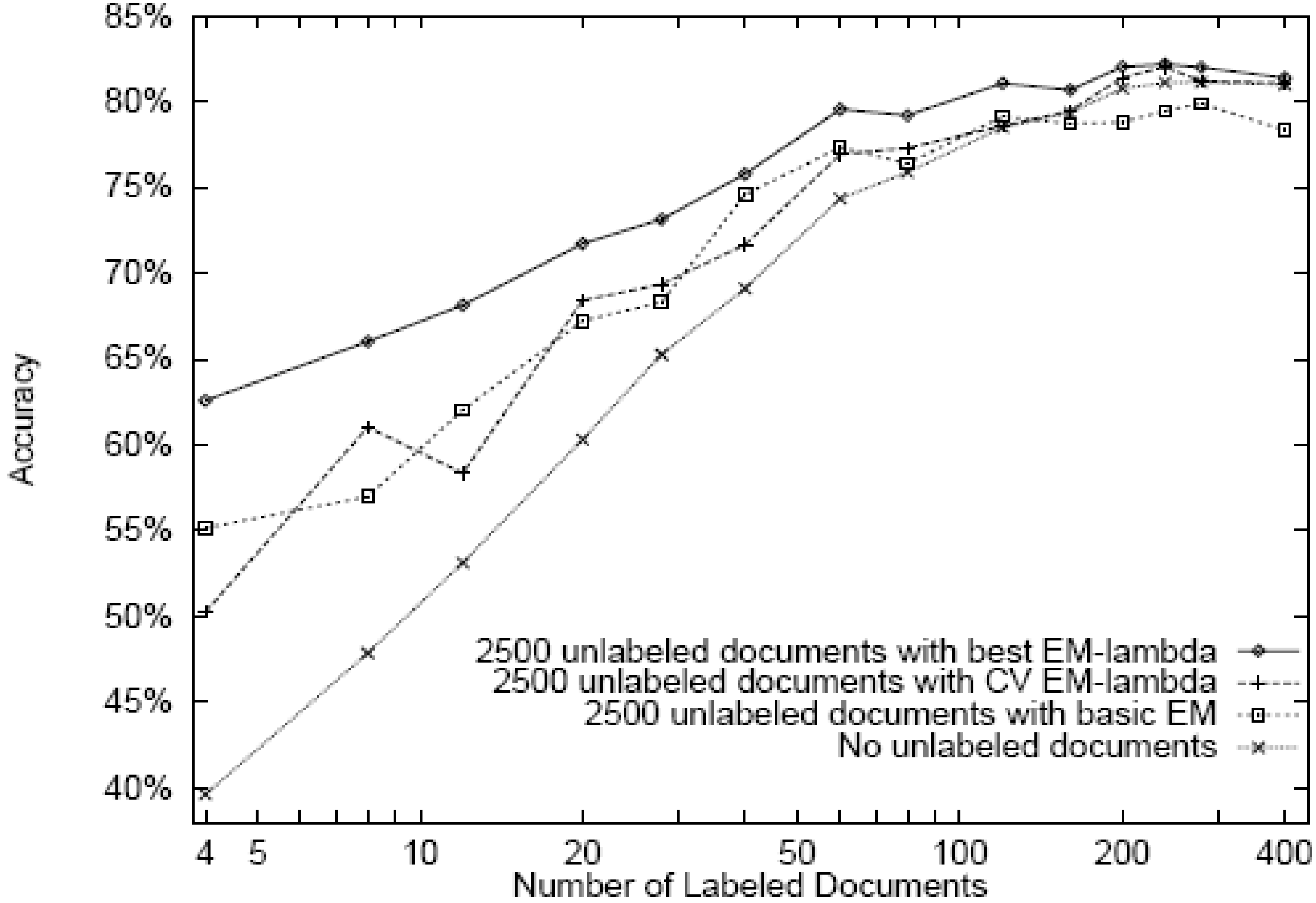


Figure 6. Classification accuracy on the WebKB data set, with modulation of the unlabeled data by the weighting factor λ. The top curve shows accuracy when using the best value of λ. In the second curve, λ is chosen by cross-validation. With small amounts of labeled data, the results are similar to basic EM; with large amounts of labeled data, the results are more accurate than basic EM. Thanks to the weighting factor, large amounts of unlabeled data no longer degrades accuracy, as it did in Figure 4, and yet the algorithm retains the large improvements with small amounts of labeled data. Note the magnified vertical axis to facilitate the comparisons.

Using Multiple Mixture Components

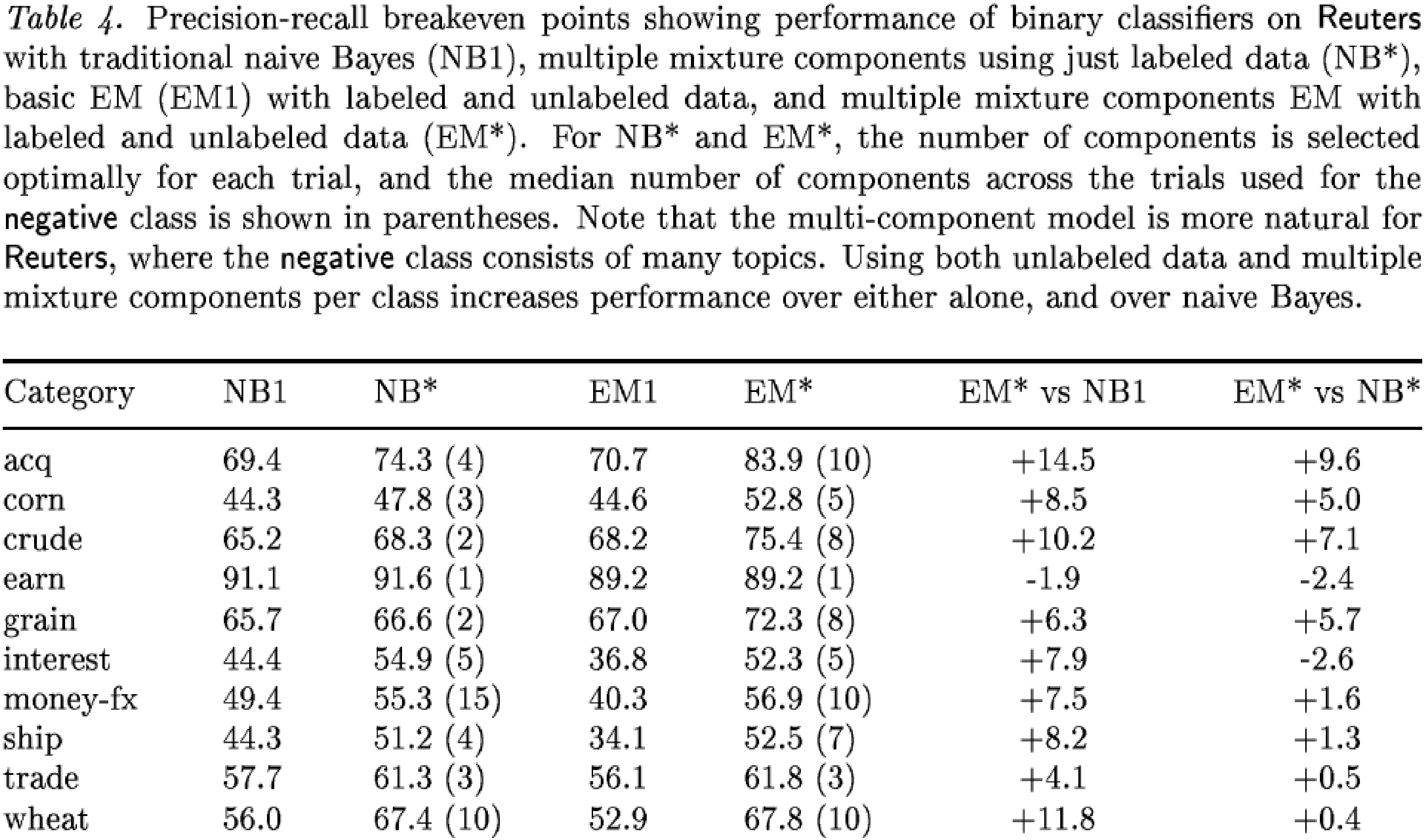


Table 4. Precision-recall breakeven points showing performance of binary classifiers on Reuters with traditional naive Bayes (NB1), multiple mixture components using just labeled data (NB*), basic EM (EM1) with labeled and unlabeled data, and multiple mixture components EM with labeled and unlabeled data (EM*). For NB* and EM*, the number of components is selected optimally for each trial, and the median number of components across the trials used for the negative class is shown in parentheses. Note that the multi-component model is more natural for Reuters, where the negative class consists of many topics. Using both unlabeled data and multiple mixture components per class increases performance over either alone, and over naive Bayes.

Category	NB1	NB*	EM1	EM*	EM* vs NB1	EM* vs NB*
acq	69.4	74.3 (4)	70.7	83.9 (10)	+14.5	+9.6
corn	44.3	47.8 (3)	44.6	52.8 (5)	+8.5	+5.0
crude	65.2	68.3 (2)	68.2	75.4 (8)	+10.2	+7.1
earn	91.1	91.6 (1)	89.2	89.2 (1)	-1.9	-2.4
grain	65.7	66.6 (2)	67.0	72.3 (8)	+6.3	+5.7
interest	44.4	54.9 (5)	36.8	52.3 (5)	+7.9	-2.6
money-fx	49.4	55.3 (15)	40.3	56.9 (10)	+7.5	+1.6
ship	44.3	51.2 (4)	34.1	52.5 (7)	+8.2	+1.3
trade	57.7	61.3 (3)	56.1	61.8 (3)	+4.1	+0.5
wheat	56.0	67.4 (10)	52.9	67.8 (10)	+11.8	+0.4

The influence of different mixture components for EM

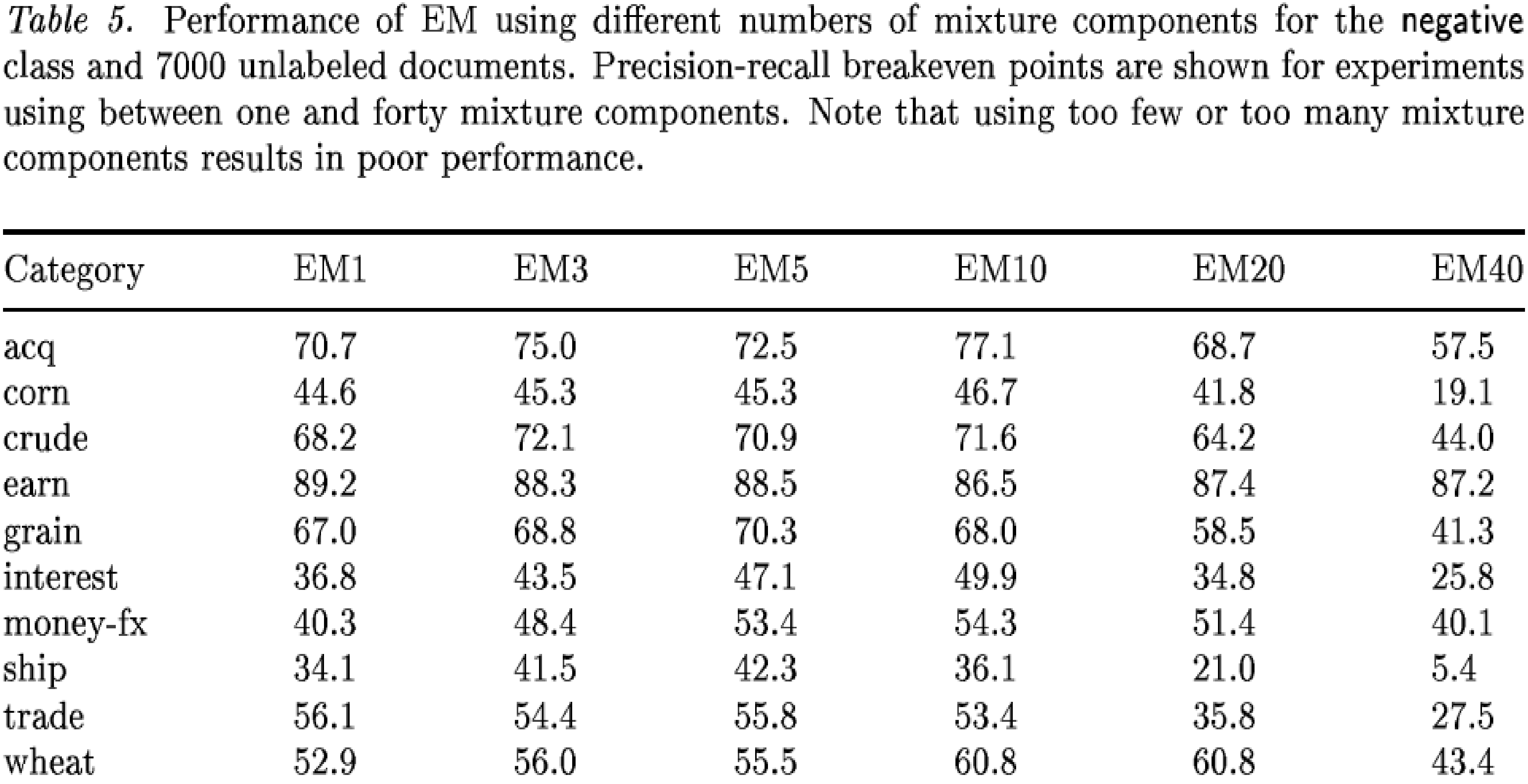


Table 5. Performance of EM using different numbers of mixture components for the negative class and 7000 unlabeled documents. Precision-recall breakeven points are shown for experiments using between one and forty mixture components. Note that using too few or too many mixture components results in poor performance.

Category	EM1	EM3	EM5	EM10	EM20	EM40
acq	70.7	75.0	72.5	77.1	68.7	57.5
corn	44.6	45.3	45.3	46.7	41.8	19.1
crude	68.2	72.1	70.9	71.6	64.2	44.0
earn	89.2	88.3	88.5	86.5	87.4	87.2
grain	67.0	68.8	70.3	68.0	58.5	41.3
interest	36.8	43.5	47.1	49.9	34.8	25.8
money-fx	40.3	48.4	53.4	54.3	51.4	40.1
ship	34.1	41.5	42.3	36.1	21.0	5.4
trade	56.1	54.4	55.8	53.4	35.8	27.5
wheat	52.9	56.0	55.5	60.8	60.8	43.4

The influence of different mixture components for NB

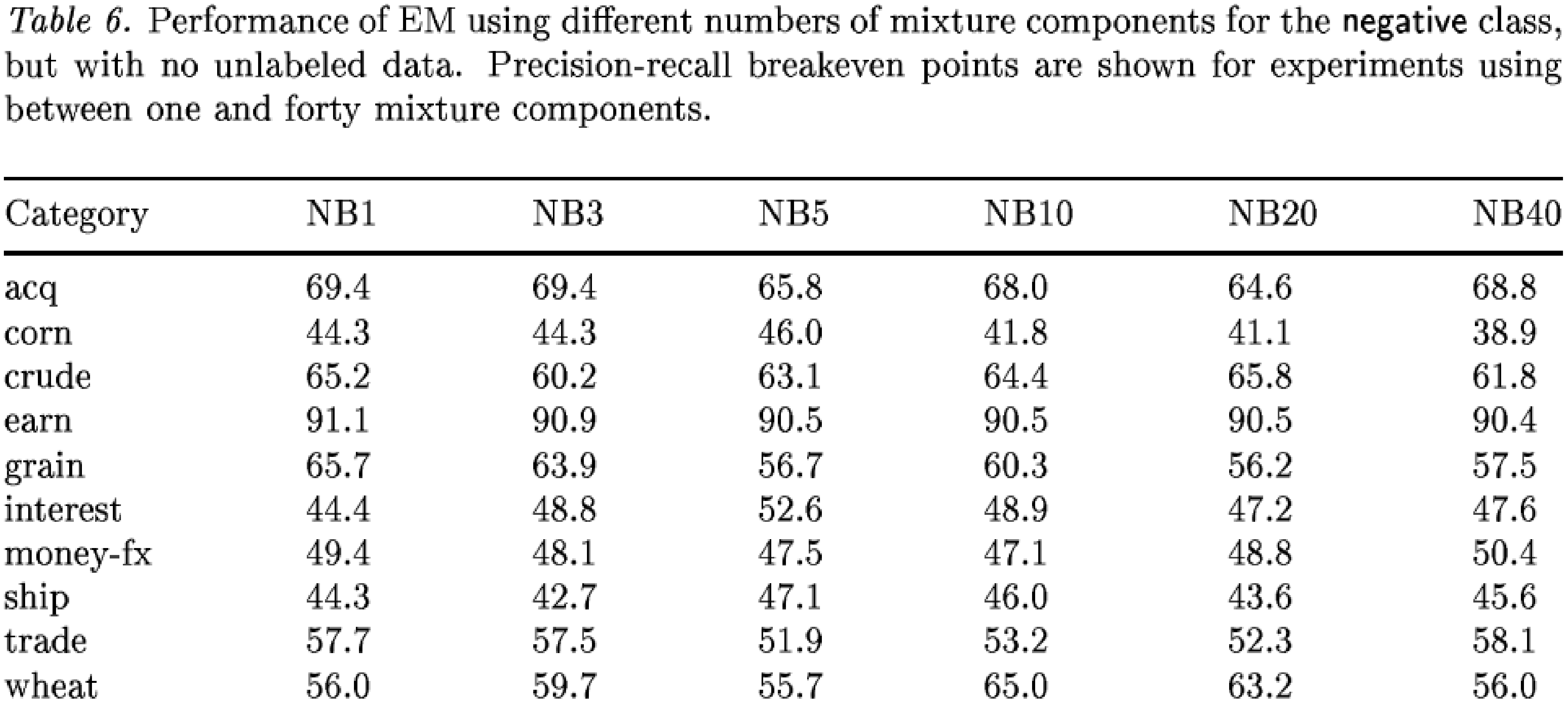


Table 6. Performance of EM using different numbers of mixture components for the negative class, but with no unlabeled data. Precision-recall breakeven points are shown for experiments using between one and forty mixture components.

Category	NB1	NB3	NB5	NB10	NB20	NB40
acq	69.4	69.4	65.8	68.0	64.6	68.8
corn	44.3	44.3	46.0	41.8	41.1	38.9
crude	65.2	60.2	63.1	64.4	65.8	61.8
earn	91.1	90.9	90.5	90.5	90.5	90.4
grain	65.7	63.9	56.7	60.3	56.2	57.5
interest	44.4	48.8	52.6	48.9	47.2	47.6
money-fx	49.4	48.1	47.5	47.1	48.8	50.4
ship	44.3	42.7	47.1	46.0	43.6	45.6
trade	57.7	57.5	51.9	53.2	52.3	58.1
wheat	56.0	59.7	55.7	65.0	63.2	56.0

The comparison of accuracy using different methods

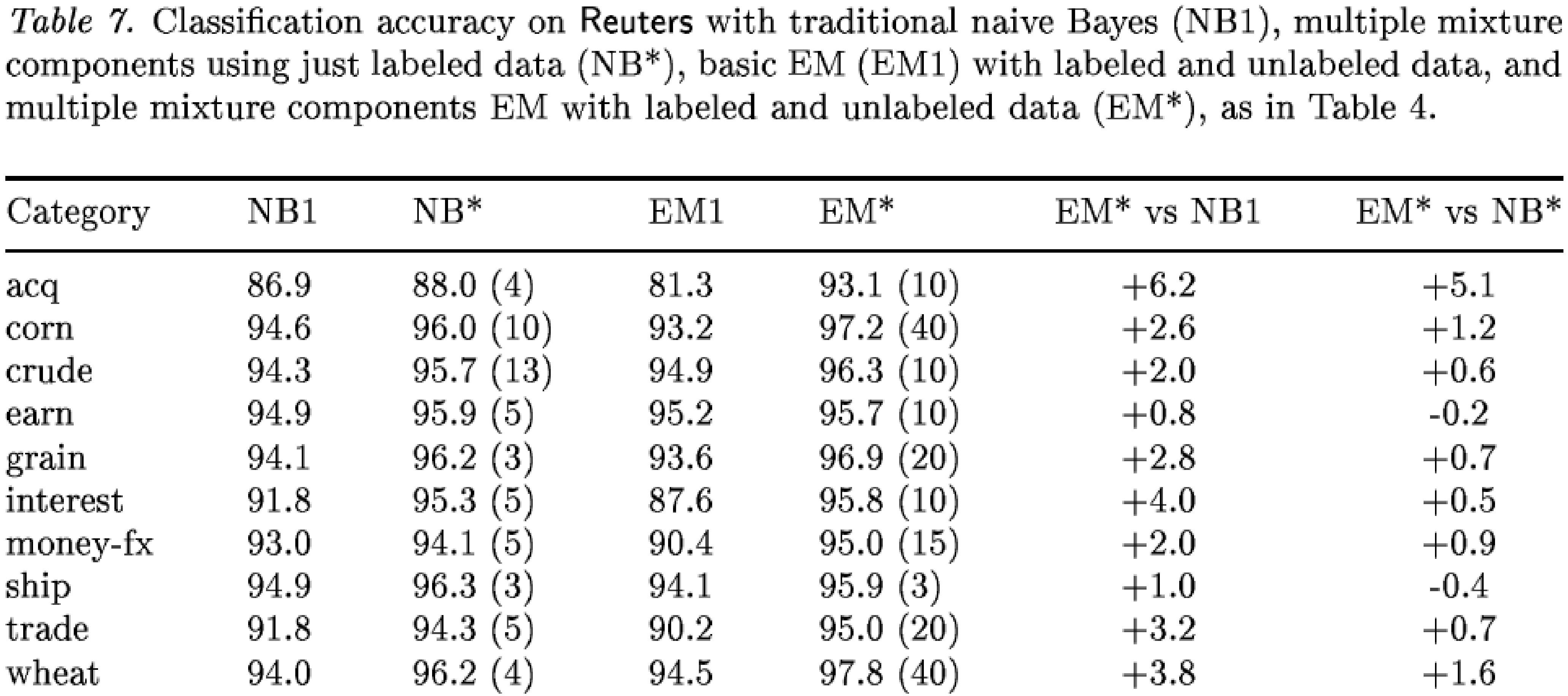


Table 7. Classification accuracy on Reuters with traditional naive Bayes (NB1), multiple mixture components using just labeled data (NB*), basic EM (EM1) with labeled and unlabeled data, and multiple mixture components EM with labeled and unlabeled data (EM*), as in Table 4.

Category	NB1	NB*	EM1	EM*	EM* vs NB1	EM* vs NB*
acq	86.9	88.0 (4)	81.3	93.1 (10)	+6.2	+5.1
corn	94.6	96.0 (10)	93.2	97.2 (40)	+2.6	+1.2
crude	94.3	95.7 (13)	94.9	96.3 (10)	+2.0	+0.6
earn	94.9	95.9 (5)	95.2	95.7 (10)	+0.8	-0.2
grain	94.1	96.2 (3)	93.6	96.9 (20)	+2.8	+0.7
interest	91.8	95.3 (5)	87.6	95.8 (10)	+4.0	+0.5
money-fx	93.0	94.1 (5)	90.4	95.0 (15)	+2.0	+0.9
ship	94.9	96.3 (3)	94.1	95.9 (3)	+1.0	-0.4
trade	91.8	94.3 (5)	90.2	95.0 (20)	+3.2	+0.7
wheat	94.0	96.2 (4)	94.5	97.8 (40)	+3.8	+1.6

The comparison of performance using different methods

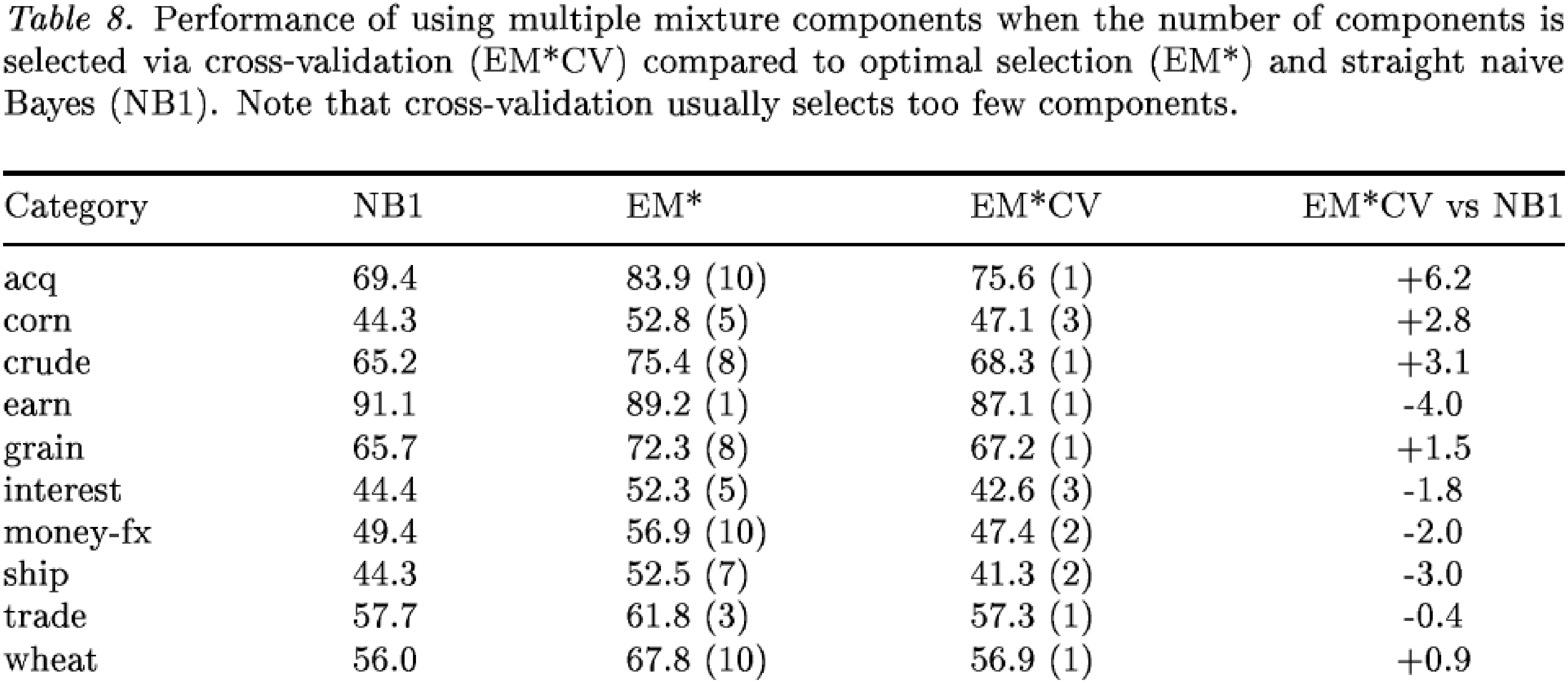


Table 8. Performance of using multiple mixture components when the number of components is selected via cross-validation (EM*CV) compared to optimal selection (EM*) and straight naive Bayes (NB1). Note that cross-validation usually selects too few components.

Category	NB1	EM*	EM*CV	EM*CV vs NB1
acq	69.4	83.9 (10)	75.6 (1)	+6.2
corn	44.3	52.8 (5)	47.1 (3)	+2.8
crude	65.2	75.4 (8)	68.3 (1)	+3.1
earn	91.1	89.2 (1)	87.1 (1)	-4.0
grain	65.7	72.3 (8)	67.2 (1)	+1.5
interest	44.4	52.3 (5)	42.6 (3)	-1.8
money-fx	49.4	56.9 (10)	47.4 (2)	-2.0
ship	44.3	52.5 (7)	41.3 (2)	-3.0
trade	57.7	61.8 (3)	57.3 (1)	-0.4
wheat	56.0	67.8 (10)	56.9 (1)	+0.9

Related Work

- EM is a well-known family of algorithms which can be implemented by treating the unclassified data as incomplete.
- It can be applied into non-textual tasks, i.e. Miller and Shahshahani, with mixture of Gaussians.
- Attention: unlabeled data does not improve the classification results in the absence of labeled data.
- EM can be combined with active-learning to improve performance.
- EM can be applied not only with naive bayes, but many machine learning algorithms like SVM, kNN.

Summary

- A family of algorithms have been presented to address the question of how to use unlabeled data to supplement scarce labeled data.
- Basic EM performs well when the data is consistent with the assumptions.
- 2 Extensions is proposed when the data is inconsistent with the assumptions.
    - EM-λ: modulating the contribution of unlabeled data.
    - Multiple Mixture Components per Class: relaxing the "one-to-one" constraint.

Questions?

Thank you!