Information Retrieval, 5, 5–40, 2002

Asymmetric Missing-data Problems: Overcoming the Lack of Negative Data in Preference Ranking

ALEKSANDER KOŁCZ ark@eas.uccs.edu
Personalogy, Inc., 24 South Weber Suite 325, Colorado Springs, CO 80903, USA

JOSHUA ALSPECTOR josh@eas.uccs.edu
Department of Electrical and Computer Engineering, University of Colorado at Colorado Springs, 1420 Austin Bluffs Pkwy., Colorado Springs, CO 80918, USA

Received February 3, 2000; Revised March 2, 2001; Accepted April 13, 2001

Abstract. In certain classification problems there is a strong a asymmetry between the number of labeled examples available for each of the classes involved. In an extreme case, there may be a complete lack of labeled data for one of the classes while, at the same time, there are adequate labeled examples for the others, accompanied by a large body of unlabeled data. Since most classification algorithms require some information about all classes involved, label estimation for the un-represented class is desired. An important representative of this group of problems is that of user interest/preference modeling where there may be a large number of examples of what the user likes with essentially no counterexamples.

Recently, there has been much interest in applying the EM algorithm to incomplete data problems in the area of text retrieval and categorization. We adapt this approach to the asymmetric case of modeling user interests in news articles, where only labeled positive training data are available, with access to a large corpus of unlabeled documents. User modeling is here equivalent to that of user-specific document ranking. EM is used in conjunction with the Naive Bayes model while its output is also utilized by a Support Vector Machine and Rocchio's technique.

Our findings demonstrate that the EM algorithm can be quite effective in modeling the negative class under a number of different initialization schemes. Although primarily just the negative training examples are needed, a natural question is whether using all of the estimated labels (i.e., positive and negative) would be more (or less) beneficial. This is important considering that, in this context, the initialization of the negative class for EM is likely not to be very accurate. Experimental results suggest that EM output should be limited to negative label estimates only.

Keywords: incomplete data problems, imbalanced training data, user modeling, personalization, information retrieval

1. Introduction

Recently, there has been an increasing interest in using unlabeled data to enhance many supervised algorithms. This, to a large extent, is due to vast amounts of "background" information that can easily (and cheaply) be obtained via the Internet, which is in contrast with the high cost and limited availability of labeled (human classified) examples. Textual domains are especially significant since, currently, the majority of Web pages contain free-text documents and text-based searches are most common. (Nigam et al. 2000) considered the

problem of enhancing classifiers with unlabeled data in text categorization where a document has to be assigned to one or more predefined categories. Typically, a training set would contain a small number of example documents for each of the classes considered, while there is also easy access to a large corpus of unlabeled documents. The Expectation Maximization (EM) algorithm (Dempster et al. 1977) was proposed as the means of assigning labels to elements of the background set. The authors showed that, indeed, EM can significantly improve the performance of a categorizer, especially if the initial number of labeled documents is small.

In the unlabeled data problems reported in the literature, it is usually assumed that there are enough "seed" documents to initialize a model for each category. Indeed, in the absence of data on the prior probabilities of the classes considered, researchers often try to make the problems as symmetric as possible, ensuring roughly equal numbers of training documents per class. However, real-life problems involve cases where information pertaining to different classes is highly asymmetric (not necessarily reflecting the true distribution of the data) and, in an extreme situation, there may be a complete lack of information for one or more classes. One important domain where such a scenario tends to occur is that of building user-preference models.

For example, in personalizing an on-line newspaper, one has only information about the articles actually visited by a user, which have to be assumed relevant unless the user is forced to provide a "negative" feedback for the documents which they did not find interesting. Indeed, a vast majority of news-personalization studies rely on some form of active user feedback in order to distinguish between the relevant and non-relevant items. Often a user is also urged to input a precise numerical rating signifying the relevance of a particular item. With lack of active feedback, one is faced with the problem of building a model having only information about what the user likes (i.e., the positive/relevant data) and no direct information about what the user dislikes (i.e., the negative/non-relevant data).

This paper addresses the problem of augmenting a ranking system for personalization of news with unlabeled documents (which will also be called *background* documents) when no negative data are present. We focus on the EM technique for obtaining label estimates for the background documents and consider several approaches of initializing the EM algorithm that are deemed appropriate for this problem setting. The effectiveness of incorporating the unlabeled-data into a user model is tested for several ranking methods including Rocchio's algorithm (Rocchio 1971), the naive Bayes (Lewis 1998) classifier and the Support Vector Machines (SVM) (Vapnik 1998). The paper is organized as follows:

In Section 2, the issues of personalization are outlined and the ranking approach to news personalization is introduced. Section 2 also outlines the related work. Section 3 discusses Rocchio's algorithm, as well as the naive Bayes and SVM architectures in the context of the ranking application. The EM algorithm and its use in the current scenario are the subject of Section 4. In Section 5, the setup of our experiments is detailed. The results are presented in Section 6 and 7, and the paper is concluded in Section 8.

2. Personalization as a special case of information retrieval

2.1. Personal news viewed as document preference ranking

Personalization of on-line news has been one of the earliest and most favored topics in the user-modeling community. Initially, the focus was mainly on USENET news (Stevens 1992), since this was the most common way of sharing large amounts of news information in the early days of the Internet. Lately, other contexts, such as on-line newspapers and fusion of different newswire feeds have been considered. Depending on the particular application, personalization can often be treated as a case of information filtering or information retrieval. In the former, for each newly appearing news item a system has to make a binary decision of whether or not to forward it to the user (i.e., accept or reject). In the latter, given a certain repository of news items a system presents them to the user employing some form of ranking, such that documents deemed to be more relevant are easier to access. Note that a practical system may need to combine both of these aspects in order to be effective.

In this work we are concerned with the information-retrieval aspect of news personalization, which we believe is more appropriate in the context of a single on-line newspaper. In such a case, the articles included in a single issue have already been filtered by the professional editorial staff and it could be harmful to the user if a personalization system were to exclude some of that information. This is particularly true if we consider that user interests may be highly variable and relying on the past information alone may ban certain current news of potential great interest. On the other hand, there could be a significant benefit if the form of presentation was modified to facilitate access to the most likely relevant articles. Indeed, existing studies demonstrate some advantage of personalizing a newspaper layout based on learned user profiles (Kamba et al. 1997).

In the variant of personalization considered here, the on-line material is assumed to be composed of sections (sports, business news, etc.), where each section consists of a list of article headlines/summaries, each pointing the user to a full version of the article. Since a user usually has no trouble identifying the general section of interest, the goal of personalization is defined as that of rank-sorting the article leads in each section such that the material likely to be more relevant to the user appears near the top of the list and is thus easier to access. In such a context, personalization is reduced to user-specific ranking of articles comprising each newspaper section. Alternative approaches to layout personalization have been proposed, but even if the layout is not directly rank-list oriented, it usually remains a function of an internal ranking of articles according to their relevance scores.

2.2. Differences from classical information retrieval

Although the view of news personalization as usually defined is close to that of classical information retrieval, there are some important characteristics affecting the process of building a user model. The usual scenario studied in text retrieval assumes a very large repository of documents and a query input facility, by which users specify their current information need. A query is usually assumed to be very short when compared to the lengths of the documents, which makes successful retrieval difficult, and several approaches (such

as query expansion and relevance feedback) have been suggested to increase the amount of information available to the system beyond what is initially given by the query. In the news-modeling scenario, the system has information related to the articles visited by the user, which can be treated as a query to the system. Such a "query" usually contains quite a wealth of information, while the number of documents in a single newspaper issue (i.e., the document repository) is much smaller than in a typical text-retrieval application. All in all, the case of news personalization can be considered as the task of information retrieval applied in a particular context, which is defined in part by the user model.

2.3. Related work

2.3.1. User modeling for personalized newspapers. Several accounts of news personalization studies have been reported in the literature. Aside from different choices of learning algorithms (which range from neural networks to Bayesian classifiers), the main differences lie in the area of their user interfaces, as well as the way user-preference information is gathered. The predominant mode is to solicit direct user feedback, such as an explicit interest profile (e.g., *NewsHound*[1] or *Crayon*[2]) or explicit rating of visited articles (e.g., (Jennings et al. 1993) or *NewsWeeder* (Lang 1995)). Combining the active feedback of one user with that of a group of several other users (i.e., collaborative filtering) has also been suggested (Claypool et al. 1999). A few studies recognize the importance of unobtrusiveness and use implicit methods to acquire user data. Morita and Shinoda (1994) measure the amount of time a user spends at reading an article and uses it to judge its relevance. *InfoScope* (Stevens 1992), a system designed for the USENET news, allows common operations such as `save`, `delete` and `reply` to affect the relevance judgement. *Anatagonomy* (Kamba et al. 1997) observes user behavior (e.g., scrolling, enlarging) during reading an article, which is then used to "modulate" the explicit user ratings. It also investigates different layout strategies based on internal scoring of all available articles.

Web browsing assistants constitute a category similar to that of news personalization systems, and in some cases the two problems are essentially the same (e.g., Billsus and Pazzani 1999). There, user interests are also learned through a profile statement (e.g., *Pointcast*[3]), explicit rating (*Syskill & Webert* (Pazzani and Billsus 1997)) or through observing user behavior while browsing (e.g., *WebWatcher* (Joachims et al. 1997) or *Letizia* (Lieberman 1995)). Since the number of Web pages available tends to be very large, the filtering aspect of personalization is of particular importance.

Unlike those studies, the work discussed here focuses on the case where there is no explicit user feedback and it is assumed that we cannot reliably obtain the non-relevant portion of the training data. Although some studies deal with this problem by considering ranking techniques based on the similarity to the relevant set only (e.g., Foltz and Dumais 1992), the ready availability of unlabeled documents prompted us to investigate the techniques of utilizing these background data with the purpose of obtaining more accurate user models.

2.3.2. Utilization of unlabeled data. Problems limited to unlabeled data only fall into the category of unsupervised learning (e.g., document clustering) and will not be discussed here. In supervised learning, Castelli and Cover (1995) showed that, in conjunction with

the unlabeled data, at least some labeled examples are generally necessary in order train an effective classifier. The case of extreme training-set imbalance addressed in this work has long been considered in the field of document retrieval, where often there is only information on the relevant class (i.e., in the form of a user's query) and no information on the non-relevant class. Croft and Harper (1979) proposed to use document/term statistics of the query and the corpus data to assign term weights and then rank the corpus documents according to their probabilistic similarity to the query. In the absence of explicitly labeled non-relevant documents, the authors suggested to initially use the overall corpus statistics (i.e., all unlabeled documents) to model the properties of the non-relevant class. The initial results can then be refined, for example via *blind feedback* (Buckley et al. 1995) (also introduced in), where the original query is used to rank the unlabeled documents, after which relevance of the top-ranked documents is assumed automatically. Such proximity based approaches (where the similarity is not necessarily derived using probabilistic arguments) have been used extensively in document ranking (e.g., see Harman 1992 for an overview)

A lot of attention has been given to problems that are not entirely one-sided, i.e., where there are (or can be obtained) some labeled examples for all of the classes. For example, "query zoning" is a recent extension of blind feedback used to select the "best" subset of non-relevant data (Mitra et al. 1997) to be used with supervised learning. One important characteristic of unlabeled data is that they are generally much cheaper to obtain than labeled examples. Hence the objective often is to make to best use of the unlabeled data to achieve high classification/ranking accuracy, while minimizing the number of labeled examples needed. In active learning (Cohn et al. 1994), an initial system is trained with a small amount of labeled data. The system then selectively inquires about the true labels for certain "problematic" elements from the unlabeled set. It has been shown that such strategies can dramatically reduce the number of labeled examples needed to achieve a certain level of accuracy (Schohn and Cohn 2000, Iyengar et al. 2000). The examples for which the label information is requested are commonly those about which the current classifier is least confident. This uncertainty can be measured either explicitly (Lewis and Gale 1994) or implicitly, by examining the outcome of several different learners and choosing the ones over which the models disagree most (Seung et al. 1992). Co-training (Blum and Mitchell 1998, Nigam and Ghani 2000) represents an effective variant of the latter approach, whereby several different models are built using different subsets of the original feature set.

One disadvantage of active learning is that it still requires an "oracle" to look up the true-label information. Nigam et al. (2000) proposed to use Expectation Maximization (EM) (Dempster et al. 1977) to estimate the class-membership probabilities of the unlabeled data and thereby to be able to incorporate them into a learning model. Clear advantages of this approach were demonstrated, especially in cases where the initial amounts of labeled examples were small. Combination of the EM approach with active learning has also been suggested (McCallum and Nigam 1997).

2.3.3. Learning with imbalanced data sets. The problems involved in dealing with imbalanced training sets (i.e., where the numbers of examples per each class are significantly different) are often encountered in practice. There are several different causes for data

imbalance. Firstly, the different numbers of elements per class may reflect the true properties of the data source. Certain events (e.g., extreme claims in insurance modeling (Pednault et al. 2000) or presence of oil spills in satellite imagery (Kubat et al. 1997)) are exceedingly rare. In medical diagnosis, certain vital measurements might also be disproportionally infrequent when compared to others (Morik et al. 2000). On the other hand, the disparity between the number of examples in different classes does not have to be linked to the original source statistics. There may be a high practical cost associated with acquiring certain types of data (e.g., in the newspaper scenario discussed here, we do not want to burden users with the task of explicitly rating every article they read), while other data (e.g., unlabeled documents) can be obtained relatively cheaply. In the field of information retrieval it has long been recognized that standard error rates used in machine learning have only limited usefulness. For example, a document collection might contain very few relevant documents, so by labeling all documents as non-relevant a retrieval system might attain a very low error rate but would also be completely useless (Yang and Liu 1999).

As summarized in Provost (2000), problems occur most often when imbalanced training data are used with algorithms applying a uniform cost function to all misclassification errors, and/or when it is automatically assumed that the class imbalance does reflect the true distribution of data. To deal with the former, it has been proposed to use asymmetric cost functions during model learning and evaluation (Pazzani et al. 1994, Domingos 1999). Also, an appropriate setting of the classification threshold can often solve the distribution-related problems (Provost 2000). Additionally, with adequate domain knowledge, the algorithms used when building a model can be adjusted to account for the inherent class imbalances and other statistical properties of the data. Pednault et al. (2000) applied such an approach to insurance risk modeling, for example.

With unmodified learners, the imbalance of the training data may have a detrimental effect on the learner's performance, depending on the nature of the task. Japkowicz (2000) analyzed this issue on a series of tasks varying in their degree of complexity, using a Multilayer Perceptron as the underlying model. The class imbalance had no effect for linearly separable problems while, in the presence of non-linearity, efforts to bring the populations of different-class examples to the same level (e.g., by oversampling the under-represented classes or subsampling the over-represented ones (Kubat and Matwin 1997)) resulted in increased performance.

3. User models for document ranking

3.1. Representation of text

Textual information is generally characterized by very high-dimensional feature spaces where the features usually correspond to words or their derivatives. To avoid some of the problems of the "curse of dimensionality" (of course word features are already high in dimensionality), models operating in the textual domain tend to ignore feature interrelationships and assume feature independence. Although such assumptions are obviously incorrect, they tend to little affect retrieval performance and, indeed, more complex systems sometimes underperform due to the inherent difficulty of accurately estimating a large

number of parameters. Sometimes a particularly simple document representation is chosen where only the binary word presence/absence attributes are considered. There is a large body of literature related to representation of text as well as to different approaches to ranking in retrieval systems. It is beyond the scope of this work to provide a thorough overview of these topics and the reader is referred to one of several comprehensive references on this subject—for example, Salton 1971, van Rijsbergen 1979, Frakes and Baeza-Yates 1992, Losee 1998.

Given a set of training documents represented as feature vectors, the goal of a user model is to be able to assign a degree-of-relevance measure to documents outside the training set. A large variety of machine learning techniques can be used for that purpose. We consider three general methodologies: *proximity-based modeling, probabilistic modeling* and *classification-based modeling* which, if considered in the stated order, are characterized by an increasing difficulty in handling the asymmetry in the availability of positive and negative training data. These three classes are exemplified, in our study, by Rocchio's algorithm, Naive Bayes, and Support Vector Machine, respectively.

3.2. Rocchio's algorithm: a proximity-based technique

In proximity-based techniques, the predominant model is that of a term-vector space, where a document is characterized by a vector of terms corresponding to words present in the document and one of several popular metrics (e.g., the normalized dot product) is used to calculate document-to-document similarity by comparing their corresponding vectors. In a retrieval event, the similarity between the query and each document in the repository is calculated and the outcome is presented as a list of documents sorted according to their similarity score. A proximity-based technique becomes the natural choice when our information about user preferences is limited to the relevant class. Given the information about non-relevant documents, more accurate feedback is possible, for example, via Rocchio's relevance-feedback algorithm (Rocchio 1971) where a document rank is determined on the basis of its "distance" to both the positive and the negative class.

Here, given the sets of relevant and non-relevant documents, $\mathcal{R}$ and $\mathcal{N}$, a reference "query" document Q is created as

$$Q = \frac{1}{|\mathcal{R}|} \sum_{d_r \in \mathcal{R}} d_r - \gamma \frac{1}{|\mathcal{N}|} \sum_{d_n \in \mathcal{N}} d_n \quad 0 \leq \gamma \leq 1 \tag{1}$$

Here, d_r and d_n denote term-weight vectors corresponding to a document from the relevant and non-relevant set, respectively, and the relevance of a new document is measured according to its distance (measured by a dot-product) to Q. The summation operator adds the document vectors term-wise, with resulting negative term-weights set to 0 by default. In this method, the term-weights in document vectors are usually given by some form of *tf-idf* (i.e., term frequency inverted document frequency) representation and the cosine distance measure is used (Harman 1992). Some enhancements of the basic Rocchio's method have recently been proposed. Dynamic Feedback Optimization (DFO) (Buckley and Salton 1995)

provides an iterative weight adaptation scheme, where the weights of Q are optimized so as to improve the ranking produced on the training set, which has been demonstrated to significantly improve the method's performance. Once the original Rocchio query is formed, the DFO adaptation process consists of several passes over the set of available terms where, during each pass, for each term it is verified whether the performance of the Rocchio technique (on the training data) would improve if the weight of the term were increased, and only those changes that lead to better performance are actually implemented. The amount by which the weights are adjusted decreases with each iteration. In their original paper, Buckley and Salton (1995) suggested three optimization passes with the weight-change rates of 0.5, 0.25, and 0.125, respectively—this setup was also adopted here (except for the case of Rocchio trained using just positive examples, where DFO was not applied). Schapire et al. (1998) showed that the enhanced Rocchio's technique performs on a text filtering task equivalently to a variant of the boosting algorithm.

In our experiments with Rocchio/DFO (except for the case of learning with positive-class data only), the following heuristic method of setting γ in (1) was chosen:

$$\gamma = \frac{|\mathcal{R}|}{2|\mathcal{N}|} \tag{2}$$

In this way the total weight contribution of the negative class is always half of the contribution of the positive class, which is motivated by our stronger confidence in the validity of the positive data. This particular choice was justified by good levels of performance obtained in preliminary experiments.

3.3. Naive Bayes: the binary independence model

In probabilistic information retrieval, the rank of a document, d_i, is directly dependent on the estimated value of its probability of relevance, $P(R \mid d_i)$, given some prior information defining the concept of relevance. Probabilistic approaches can, in principle, be quite complex by attempting to model dependences between words in a language. Surprisingly, however, extreme simplifications often tend to produce very good results in practice. In particular, a simple probabilistic model commonly known as naive Bayes has found numerous successful applications (Lewis 1998).

A Bayesian network explicitly models the class-conditional probabilities and uses Bayes theorem to estimate the probability of class membership given the input data. In a multi-category scenario, the quantities being estimated are the posterior probabilities $P(c_j \mid d_i;\theta)$, where c_j denotes one of the classes, d_i is a document and θ is a vector capturing the parameters of a model. To simplify the notation, in the current two-class scenario the classes will be denoted as R (relevant documents) and N (non-relevant documents) and, without loss of generality, the parameter vector θ will be omitted. Thus the point is to estimate $P(R \mid d_i)$, using the Bayes theorem, so that it can be used as a degree-of-relevance measure for a document d_i. Although this process can potentially be very complex, a simplistic approach is to consider only the binary presence/absence of terms in a document and assume conditional independence of word features. In the resulting binary independence

model (BIM—also known as the multivariate-Bernoulli model (Robertson and Sparck-Jones 1976, McCallum and Nigam 1998), the class-conditional probability of a document is given by the following formula, where a document d_i is represented by an $|\mathcal{V}|$-element term vector in the context of a dictionary $\mathcal{V}$:

$$P(d_i \mid R) == \prod_{\omega_t \in \mathcal{V}} B_{it}\, p(\omega_t \mid R) + (1 - B_{it})\, p(\omega_t \mid N) \tag{3}$$

where, B_{it} equals 1 if document d_i contains term ω_t and is 0 otherwise. Note that both the presence and absence of individual terms is taken into account (i.e., regardless of a document's real length, $P(d_i \mid R)$ is the product of exactly $|\mathcal{V}|$ terms, where $|\mathcal{V}|$ is the size of the dictionary), while the frequency of terms within a document is ignored. If $\mathcal{B}_i$ denotes the set of terms present in document d_i, formula (3) can be expressed as:

$$P(d_i \mid R) == \prod_{\omega_t \in \mathcal{B}_i} \frac{p(\omega_t \mid R)}{1 - p(\omega_t \mid R)} \prod_{\omega_t \in \mathcal{V}} (1 - p(\omega_t \mid R)) \tag{4}$$

In the binary independence model, the relevance probability of a term is estimated as (McCallum and Nigam 1998):

$$p(\omega_t \mid R) = \frac{1 + \sum_i B_{it} P(R \mid d_i)}{2 + \sum_i P(R \mid d_i)} \tag{5}$$

where i indexes the available documents. The above formula features Laplace smoothing to prevent rare terms from having the probability estimates of 0. If document ranking is to be performed according to $P(R \mid d_i)$, this probability is calculated using the Bayes theorem as:

$$P(R \mid d_i) = P(R) \frac{P(d_i \mid R)}{P(d_i)} = P(R) \prod_{\omega_t \in \mathcal{B}_i} \frac{p(\omega_t \mid R)}{p(\omega_t)} \prod_{\omega_t \in \mathcal{V}} \frac{1 - p(\omega_t \mid R)}{1 - p(\omega_t)} \tag{6}$$

where the first product is taken over the terms appearing in the document d_i, and the second product is taken over all terms in the dictionary, $\mathcal{V}$. To compare $P(R \mid d_i)$ for different documents, the common terms in (6) can be eliminated, leading to:

$$P(R \mid d_i) \propto \prod_{\omega_t \in \mathcal{B}_i} \frac{p(\omega_t \mid R)}{p(\omega_t)} \propto \sum_{\omega_t \in \mathcal{B}_i} \log \frac{p(\omega_t \mid R)}{p(\omega_t)} \tag{7}$$

Note that, in a user-modeling scenario, the labeled positive data can be used to estimate $p(\omega_t \mid R)$ for each term, while the unlabeled "background" data can be taken to represent both the relevant and non-relevant documents and thus be used to estimate $p(\omega_t)$ for each term. Therefore, in principle, naive Bayes does not require labeled negative data, but better empirical performance tends to be achieved for a decision-theoretic measure of relevance

(Losee 1998) based on the odds $P(R \mid d_i)/P(N \mid d_i)$ (Robertson and Sparck-Jones 1976) rather than on $P(R \mid d_i)$ alone. With such a measure, a document rank is a function of:

$$\frac{P(R \mid d_i)}{P(N \mid d_i)} \propto \prod_{\omega_t \in \mathcal{B}_i} \frac{p(\omega_t \mid R)}{p(\omega_t \mid N)} \propto \sum_{\omega_t \in \mathcal{B}_i} \log \frac{p(\omega_t \mid R)}{p(\omega_t \mid N)} \tag{8}$$

Note that examples of negative documents are needed to obtain the estimates of $p(\omega_t \mid N)$. It has been argued that since typically the majority of documents is likely to be not relevant to a user/query, formulas (7) and (8) should result in close levels of performance (i.e., $p(\omega_t \mid N) \approx p(\omega_t)$). The decision-theoretic measure (8) has often been proved superior in practice, however, and it is not clear if the non-relevance assumption about background documents would hold in a limited-data environment such as an on-line newspaper.

3.4. Support Vector Machines

Support Vector Machines (SVM) (Vapnik 1998) represent a new class of machine-learning algorithms that explicitly estimate the location of the inter-class boundary. SVMs have been shown to be very effective in many classification tasks, including text categorization (Joachims 1997, Yang and Liu 1999). In the classification setting, an SVM transforms the input domain into a possibly infinite dimensional space, in which the classification boundary can be modeled as a hyperplane. To identify the best hyperplane, an SVM finds a set of training points that directly determine the classification-error margin and ignores the rest of the training set. The chosen points are known as support vectors. In particular, given a set of linearly-separable points (which is often the case for textual data) $\{\mathbf{x}, y : y = \pm 1\, \mathbf{x} \in \Re^D\}$, an SVM is defined by a hyperplane for which the inequality $y\,(\mathbf{w} \cdot \mathbf{x} + b) \geq 1$ is satisfied by all elements of the training set and the norm of the hyperplane normal vector, $\mathbf{w}$, is minimal. In some variants of SVMs the dot-product $\mathbf{w} \cdot \mathbf{x}$, can be substituted by an expression involving a nonlinear kernel function, which is useful if the original dimensionality of the input space is low. In text-domain applications, the word-feature space has inherently very high dimensionality and such transformations are rarely necessary. In our experiments we considered a linear-kernel SVM and used the *SVM^{light}* package[4] written by Thorsten Joachims (Joachims 1999).

Note that unlike the case of naive Bayes, an SVM does not explicitly estimate the class-membership probabilities. In the ranking application, the goal is not so much to classify the points as to sort them according to their degree of relevance which, in the case of an SVM, can be measured by their distance to the separating hyperplane. To this end, the actual position of the hyperplane is less relevant than its spatial orientation although both of them are determined by the choice of support vectors. The presumed relationship between the output of an SVM and the degree of relevance of a document is somewhat arbitrary. However, Platt (2000) has recently shown that by appropriately post-processing the SVM outputs, one can obtain a reliable estimate of the confidence measure in the classification decision, which supports the methodology chosen in this work.

An SVM clearly requires that examples of both classes are provided by the training set. Because the position and orientation of the separating hyperplane are determined by the

most similar elements of the positive and negative training sets, the choice of the negative examples is critical for this type of architecture.

4. Obtaining label estimates with EM/naive Bayes

4.1. The baseline assumption: is negative labeling really necessary?

Of course, given positive training data only it is still possible to create a preference ranking, for example by using a proximity-based technique. The research in information retrieval has shown however that, by asking users to explicitly rate the relevance of the documents retrieved, significant performance gains can be made. In the absence of such active feedback, one might take a seemingly reasonable *baseline* assumption that the majority of unlabeled documents are, in fact, not relevant to the user. In information retrieval where large document repositories are searched with narrowly defined user queries, such an assumption may be well founded. In the newspaper personalization scenario, however, assuming automatic non-relevance of articles not visited by the user may be unjustified for several reasons. Users usually do not have enough time to examine all articles in a single newspaper issue, so they may well miss many relevant ones. The time constraint may force a user to prioritize information so that only the most important articles are visited leaving others unexplored, although users may wish they had time to explore them. Also, because a user's interests tend to be vaguely defined, a higher proportion of the documents available may be of interest, when compared with the usual retrieval scenario (e.g., a search engine). Finally, there may be a substantial overlap between different news stories and, once a user learns the information from one of them, the others may be ignored although, initially, all of them represented similar levels of relevance. Some authors tackle the latter problem by separately modeling users' short-term and long-term interests (Billsus and Pazzani 1999). All in all, it appears that the baseline assumption about the non-relevance of background documents may be prone to error, the effects of which are likely to be more profound in the case where the unlabeled data outweigh the labeled data. The baseline assumption should nevertheless be useful in evaluating the effectiveness of alternative approaches. In the experiments discussed later in this paper the baseline assumption is used to obtain a reference level of performance for each of the ranking methods considered.

4.2. Estimating the missing labels with EM

Given labeled training data and a body of unlabeled documents, the EM algorithm (Dempster et al. 1977) provides a means of estimating the missing labels. The EM algorithm is a general iterative procedure designed to obtain maximum-likelihood parameter estimates in incomplete-data environments. Nigam et al. (2000) presented an extensive analysis of the use of EM in the context of document categorization where the missing values are the class labels of subsets of documents (note that EM can be applied in other important incomplete-data problems (McLachlan and Krishnan 1996)). In particular, the authors combined the EM procedure with the multinomial naive Bayes model (see below) and demonstrated

significant improvements in classification accuracy with their approach. Their promising results motivated us to adapt this technique to the current problem. Below, a brief description of the EM application is given.

It is assumed that documents are distributed according to a mixture distribution, where mixture components correspond to distinct class labels. In the two-class scenario considered here this can be expressed as:

$$P(d_i \mid \theta) = P(R \mid \theta)P(d_i \mid R; \theta) + P(N \mid \theta)P(d_i \mid N; \theta)$$

where d_i is a document, R and N represent the classes of relevant and non-relevant documents, respectively, and θ is a vector concatenating the parameters for relevant and non-relevant class models. Given a particular parametric model, its initial settings, and a corpus of labeled and unlabeled data, the EM algorithm executes a sequence of two-step iterations until convergence. In the *(E)xpectation* step, the current parameter values are used to estimate the class-membership probabilities of unlabeled data, while in the *(M)aximization* step, the values of model parameters are re-computed using the fixed labels, as well as the class-membership probability estimates through an application of the maximum likelihood principle.

There are many categories of incomplete data problems in which the above formulation would have to be modified to fit a particular problem setting. In the formulation given above, the EM procedure is inherently tied to a particular parametric model, and its ultimate purpose is to utilize the unlabeled data to improve the model over its initial settings. Thus the assignment of class-membership probability estimates to the unlabeled training set may be considered to be a by-product of the parameter estimation process. In principle, if EM were to be used in conjunction with several parametric models, a separate procedure should be devised for each one, using a suitable maximum likelihood estimator for each model's parameters. This would be more or less difficult depending on the characteristics of the model. We consider a simplified approach, whereby only one model (i.e., the naive Bayes) is used in conjunction with the EM to arrive at the class-membership probabilities of the unlabeled data, and these results are then re-used by all the models considered. Note that in this case the process of obtaining the label estimates is no longer just a by-product of the EM but, instead, is treated as its primary outcome. The choice of the model (i.e., the naive Bayes) used with the EM certainly biases the results. However, the use of naive Bayes in conjunction with EM has been proven to be very effective in prior research (Nigam et al. 2000), and since the class-membership probabilities assigned to unlabeled data in such a setup tend to be close to the extremal 0, 1 values, the technique should prove useful in the context considered here.

4.2.1. The multinomial document event model. In their experiments with EM/naive Bayes, Nigam et al. (2000) considered a multinomial generative document model which leads to a version of the naive Bayes classifier, alternative to BIM, where only the terms present in a document are taken into account (i.e., documents of different lengths will have different numbers of product terms in their probabilistic representation). Also, the frequency

of terms within a document is accounted for. Their choice was motivated by prior research (McCallum and Nigam 1998), where the categorization performance of the multinomial and multivariate-Bernoulli variants of naive Bayes was compared and an advantage of the multinomial model was demonstrated. In the multinomial event model $P(d_i \mid R)$ is computed as

$$P(d_i \mid R) \propto \prod_{\omega_t \in d_i} p(\omega_t \mid R) \tag{9}$$

with $p(\omega_t \mid R)$ estimated as (McCallum and Nigam 1998):

$$p(\omega_t \mid R) = \frac{1 + \sum_i N_{it} P(R \mid d_i)}{|\mathcal{V}| + \sum_t \sum_i N_{it} P(R \mid d_i)} \tag{10}$$

where $|\mathcal{V}|$ is the dictionary size, N_{it} is the number of occurrences of term ω_t in document d_i and $P(R \mid d_i)$ is the probability of relevance of document d_i (for labeled documents $P(R \mid d_i)$ is either 0 or 1).

Although appropriate for classification, the multinomial model is not suitable for ranking applications, since the relevance values produced are not normalized with respect to the length of the documents, and thus short and long documents may be assigned significantly different relevance values (Lewis 1998). This is because, in the multinomial model, the length of a document's feature vector determines the number of terms in its probabilistic representation (9). The binary independence model (BIM), on the other hand, always takes all possible features (whether present or absent) into account, which results in fixed-length feature vectors regardless of the actual length of the documents being considered.

This would make the multivariate-Bernoulli model a natural choice for use with the EM procedure in the current context. Our preliminary experiments showed, however, that such a combination is not appropriate if there is an asymmetry between the number of labeled training documents available for the individual classes. We observed that the class having more labeled documents tends to dominate the EM process, resulting in most of the unlabeled documents being eventually assigned to that class. This behavior is largely unaffected by initializing the unlabeled documents to different class-membership probabilities even if these probabilities favor the class which is under-represented. This is clearly undesirable in the context of the current problem since, by definition, there are no hard-labeled negative examples at our disposal.

A possible explanation for such a behavior is that, in the BIM naive Bayes, all terms in a dictionary are always taken into account. The probability of terms corresponding to words absent in a document are always accounted for and are non-zero, so all documents and all words have initially some non-zero probability of belonging to the strongly represented classes. This, in turn, strengthens the probabilities of words in these documents. At the same time, the overall probability of the strongly represented classes is not allowed to drop due to the presence of many fixed-label examples for those classes. This stabilization is much

weaker for the under-represented class, which causes a gradual decrease of its probability as the EM process continues.

The multinomial model, on the other hand, appears not to suffer such from such effects and is much more stable under different modes of EM initialization. To illustrate this point we performed a two-class experiment where the number of fixed positive documents was held constant, while the number of fixed negative documents was varied such that the ratio of the numbers of fixed-label negative-to-positive documents took values in $\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, 0\}$. At the same time, before the commencement of the EM process, the relevance probabilities of the unlabeled documents (UL) were uniformly initialized to $P(R \mid UL) \in \{\frac{1}{2}, \frac{1}{4}, \frac{1}{10}\}$. The EM process was run till convergence and the final proportion of documents labeled as positive was measured (note that this number does not include documents that were assigned fixed labels at the outset). The data used in this experiment corresponded to documents which a group of 6 users found relevant/non-relevant—a detailed description of the dataset is given in Section 5.1.1. The results (averaged across the user group) for the binary-independence and multinomial models are presented in Tables 1 and 2, respectively. We can see that when the asymmetry between the number of fixed-label documents increases, EM with BIM tends to assign unlabeled documents to the most represented (i.e., positive) class no matter what their initialization value. EM with the multinomial model, on the other hand, although still sensitive to the number of fixed-label documents, is much more stable and strongly depends on the initialization value given to the unlabeled documents.

Taking the above into consideration, we chose the multinomial model to be used in conjunction with EM to estimate the relevance probabilities of the unlabeled documents. These were then utilized by the multivariate-Bernoulli naive Bayes and other techniques considered to obtain the document ranking.

Table 1. Results for the BIM-EM experiment. When the proportion of labeled documents increasingly favors one class (the R class in this case), EM tends to assign most unlabeled documents to that class, even if their initial class-membership probabilites favor the other class.

	Ratio = 1	Ratio = 0.5	Ratio = 0.25	Ratio = 0.125	Ratio = 0
$P(R \mid UL) = 0.5$	0.69	0.78	0.85	0.95	1
$P(R \mid UL) = 0.25$	0.26	0.43	0.58	0.86	1
$P(R \mid UL) = 0.1$	0.18	0.29	0.47	0.85	1

Table 2. Results for the multinomial-EM experiment. EM can assign a significant portion of the unlabeled documents to the under-represented class, depending on the initial class-membership probabilities assigned to the unlabeled set.

	Ratio = 1	Ratio = 0.5	Ratio = 0.25	Ratio = 0.125	Ratio = 0
$P(R \mid UL) = 0.5$	0.56	0.46	0.39	0.35	0.31
$P(R \mid UL) = 0.25$	0.25	0.24	0.23	0.23	0.23
$P(R \mid UL) = 0.1$	0.20	0.20	0.20	0.20	0.21

4.3. Negative class initialization for EM: several approaches

Given our general lack of information about the negative document class, initialization of the EM process must necessarily be somewhat ambiguous. The question addressed by this paper is whether the EM procedure can produce beneficial results under these asymmetric conditions and whether there is a clear advantage of using one mode of initialization over another. Normally, if labeled examples for each class were actually available, initialization of the unlabeled data would not be necessary. In such a case, an initial model would be built using just the labeled examples and then, in the EM loop, this model would be applied to estimate the class-membership probabilities of the unlabeled data, which in turn would be used, in conjunction with the labeled examples, to arrive at new estimates of the model parameters, and so on. If such an approach were to be applied in the current scenario, it would lead to the first (i.e., during the first EM iteration) estimates of *non-relevance* probability for the unlabeled documents to be close to 0. Recall, however, that it is reasonable to assume that most of these documents are in fact non-relevant and it appears desirable to use this prior information to initialize the EM process.

Prior to starting the EM process we therefore propose to use an extra E-step where the class-membership probabilities of the unlabeled data are set according to our assumptions about their non-relevance, and an extra M-step where both the labeled data and thus initialized unlabeled data are used to set the initial model parameters. In general, several ways of initializing the labeling process can be considered (for future reference we designate them with terms shown in triangular brackets):

1. <const-init> initialize all unlabeled data as members of the negative class (i.e., $P(R \mid UL) = 0$) and allow their labels to be settled by EM. This is an extension of the baseline assumption, which is reasonable in the sense that most documents in a large corpus are likely not to be relevant to a particular user.
2. <proximity-init> use a proximity-based technique to initially rank all unlabeled documents according to their similarity to the positive class; subsequently use these ranks to assign initial probabilities of relevance to all unlabeled documents and allow them to be settled by EM. More precisely, if i indicates the position of unlabeled document in the ranked list (where for the top ranking document $i = 1$), then its probability of relevance is initialized as $0.5/i$.
3. use a proximity-based technique to identify the set of K unlabeled documents that are least similar to the positive class (we chose the value of K to be equal to the number of labeled positive documents). Label them as negative while leaving the rest of the background documents unlabeled and then run the EM. Two variants of this initialization procedure can be considered. In one <tail-free-init>, the initially labeled documents are allowed to have their class membership probabilities modified by the EM process while in the other <tail-clamp-init> variant, the initial assignment of the K negative labels remains fixed throughout the EM process. The rationale for fixing the initial labels is stabilization of the EM procedure considering that the labels of the positive training set also remain fixed. Note that this approach is related to blind

feedback technique of information retrieval (Buckley et al. 1995), where the original query is used to identify a set of top ranking documents from the unlabeled set to be then treated as relevant. Our technique is similar, but the bottom ranking documents are treated as non-relevant instead.

In all cases, documents d_i in the positive training set have their relevance probabilities, $P(R \mid d_i)$, fixed at 1. Once the EM process is run till convergence, we are given the class-membership probabilities for all elements of the unlabeled data set. A question arises as to whether all of them should be further used in training a model to be used for ranking. In Nigam et al. (2000), the authors point out that the assumptions taken by EM are often violated by real data, in which case the use of EM may in fact hinder the performance of the model. In particular, data may not be accurately modeled by the mixture distribution chosen and/or the mixture components may not correspond to class labels. In the case of user modeling for personalizing the news, it is almost certain that both the classes of relevant and non-relevant documents will themselves be represented as mixtures over many different and/or overlapping topics. Also, in some cases the relevance/non-relevance boundary may "cut" across a single mixture component (e.g., a user may be interested in certain aspects of international affairs but not in others). Thus we can have only limited confidence in the EM outcomes.

Recall that the rationale for performing EM in the first place was the lack of negative examples while the positive training set was deemed adequate. One could thus use just the portion of the unlabeled data for which $P(N \mid d) > P(R \mid d)$, while ignoring the rest. This approach seems reasonable since the EM process, followed by a process of converting the $P(R \mid d)$ probability estimates to class labels, may result in mislabeling some documents as positive thus introducing "noise" into the positive class. On the other hand, within the set of background documents labeled as positive as a result of EM, the majority might correspond to the correct labeling, in which case utilizing the complete labeling results would be more beneficial. Since there is no clear rationale for using either approach we compare both of them in our experiments.

5. Experimental setup

5.1. Data acquisition

5.1.1. The newspaper experiment (real users). The data used in our real-newspaper experiment were acquired by means of a personalization system developed at the University of Colorado, which was based on the on-line version of the New York Times (NYT). The users were offered a browsing interface essentially identical with that of the NYT website and were asked to visit a few articles a day. The users were asked to simply read the newspaper as normal and were not forced to read any particular number of documents or to change their usual interaction pattern with NYT website. For data collection, no personalization was offered to the users, in order not to skew the subsequent results (i.e., each user saw the unmodified NYT website). During the browsing process, a click on an article resulted in storing the lead (i.e., the headline/summary combination) of that article

in the user's profile, together with the URL and timestamp information. Additionally, a comprehensive set of articles accessible from the newspaper's site for each particular day was recorded (i.e., the lead material of those articles was stored). This constituted a corpus of 20,000 documents which, on average (the standard deviation is given using a $\pm$ notation), contained 5 ± 3 words per headline and 32 ± 10 words per summary (the corresponding numbers of unique words, were 5 ± 2 and 29 ± 8, respectively). Over the period of 3 months (ending in June 1999) about 40 users (mostly students) registered at the site, out of which a group of the top 6 users was chosen for analysis, where each member of the group visited approximately 100 or more articles. The actual numbers of visited articles were: 91, 99, 107, 150, 158 and 299. During each session with the system, a user would typically click on 0–4 articles, mostly from the NYT front page (which provides links to 40–60 articles). The total number of articles that a user could have potentially clicked on varied from session to session and was in the range of a few hundred (the actual numbers varied and depended on they day of the week). Only the headline-summary combination was used to represent an article in a user's profile, which averaged to approx. 37 words (31 unique) per article. Without loss of generality, we will refer to the lead descriptors as *documents*.

For each user, the profile information was chronologically ordered and split such that the initial 60% of articles were used to build a model, while the rest were used for testing. These data will be referred to as the positive training set and positive testing set, respectively. The positive training data for each user were augmented by a random sample of unlabeled documents taken from the same time window as the positive training set. For each user, the sets of positive and unlabeled documents were disjoint, but it was not known if the user actually had a chance to see any of the unlabeled documents. To examine the effects of the amount of unlabeled data on the ranking performance, we considered seven cases, where the ratio of the number of unlabeled documents to the number of positive training documents was equal to $ratio = 1/8, 1/4, 1/2, 1, 2, 4$ and 8.

To generate the negative test set, each user was asked to identify the relevant documents (i.e., those that each particular user thought they might have clicked on, given the chance) in a random pool of 300 (selected out of 20,000) documents corresponding to the same time window as the positive test set. The complement of this set was then treated as the negative data, while the documents judged as relevant in this process were ignored and not used in the experiments below. A random sub-sample of the negative data was selected so that, for each user, the numbers of the positive and negative test documents were equal.

5.1.2. The Reuters experiment (synthetic users). Since the 6-individual user set might be considered rather small, we chose to validate our results with a synthetic-user setup, where the data could be more readily obtained. We chose the Reuters-21578 dataset (Mod-Apte split[5]), which has been studied extensively in the text categorization literature. The collection consists of a number of Reuters newswire articles in 93 categories, and is divided into a 6903-document training set and 3299-document test set. Only the 90 categories containing at least one training and at least one test document were considered. Based on these data,

a collection of 100 synthetic users was generated, where each user's interest was modeled through a mixture distribution over different topics by assigning a relevance weight to each category. This was conducted as follows:

1. The Reuters categories were ranked according to the number of documents per category (taking both the training and the test corpora into consideration). The top 20 categories were selected as the ones for which non-zero relevance weights could be chosen.
2. For each user, the following set of relevance (mixing) weights was randomly distributed among the top 20 categories:

$$5, 4, 4, 3, 3, 3, 2, 2, 2, 2, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0$$

   with the remaining 80 categories being assigned the weight of 0. This mixture distribution was chosen to approximate (in a very rough way) the interest profile of a real person, who is likely to be very interested in a fairly small number of topics, have a moderate number of general interests, and be indifferent to a large number of specialized topics.
3. Given this distribution, a positive training set for a user was generated using probabilistic sampling without replacement out of the Reuters training corpus, where only documents whose categories had non-zero relevance weights were eligible for selection. A corresponding process was used to generate a positive test set. The negative test set was generated by uniform sampling from the categories assigned the weight of 0, and then by a uniform sampling from the test documents available for the selected category. The unlabeled set (based on Reuters training corpus) was generated by uniform sampling from all categories, and then by a uniform sampling from the training documents available for the selected category.
4. For each user, the positive training set contained 200 documents, while the test set contained 100 positive and 100 negative documents. As in the newspaper experiment, the number of unlabeled documents was varied in proportion to the size of the positive training set. With the ratio of the number of unlabeled documents to the number of positive training documents taking values from the set $\{1/8,\ 1/4,\ 1/2,\ 1,\ 2,\ 4,\ 8\}$. The number of unlabeled documents ranged from 25 to 1600.

It has to be stressed that the simulated-user dataset based on the Reuters corpus differs in several important ways from the real user data used in the newspaper experiment. In the case of Reuters data, all users share the same distribution of interests across different topics and the diversity of topics does not necessarily depict a genuine scenario. Also, within each topic the documents are treated uniformly during sampling, which does not correspond to realistic user behavior (e.g., while in the business section of a newspaper people are not likely to read articles at random). Nevertheless, this simplistic setup attempts to capture the basic statistics of a user interest profile.

5.2. *Document representation*

For each document, an initial filtering stage removed all non-alphanumeric characters and the remaining strings of contiguous non-whitespace characters were treated as words. Each document was then represented by a vector of terms, where a term corresponds to a stemmed lowercase version of a word (using the algorithm due to Porter (1980)). The use of stemming was motivated by the general scarceness of the lead data (in the newspaper experiment), where stemming should lead to an increased term frequency. A stop-list was implemented to eliminate common/irrelevant terms but no further feature-space reduction was used. The use of term and document frequency information was algorithm specific. The Rocchio technique used standard *tfidf* weighting, while the multinomial naive Bayes (used with the EM algorithm) took just the frequency of terms in documents into account. In the case of the naive Bayes/BIM and the SVM, each document was encoded as a binary feature vector, where features corresponded to the presence/absence of a particular term in the document.

5.3. *Measurement of performance*

The information retrieval literature often uses the traditional notions of *precision* and *recall* to measure a system's performance. In a retrieval scenario, precision is the ratio of the number of relevant documents retrieved to the total number of documents retrieved, while recall is defined as the ratio of the number of relevant documents retrieved to the total number of relevant documents in the collection. In the current scenario, these "raw" measures are not directly applicable since, in a single list context, all elements of the list are visible to the user so recall is always 100% and the precision is constant regardless of list ordering. To overcome this difficulty, average non-interpolated precision is used, defined as (Schapire et al. 1998)

$$AvgP = \sum_{d \in \mathcal{R}} \frac{\text{number of relevant documents } d_j \text{ for which } rank\,(d_j) \leq rank\,(d)}{rank(d)} \quad (11)$$

where $\mathcal{R}$ is the set of relevant documents. Here the precision of a system is averaged for relevance threshold values that cause a change in the number of relevant documents returned by the system (i.e., we analyze what would be the system's precision if a hard-threshold filtering decision was made about the relevance of each document for a range of threshold values).

Note that with the average non-interpolated precision chosen as the performance metric, a random ordering of the test-set documents would result in $AvgP$ of 50% while a "perfect" ordering would produce $AvgP$ of 100%.

6. Results (newspaper experiment)

All results presented below refer to averaging across the 6-element subject set. Notation $m \pm std\_dev$ is used to denote the mean and standard deviation across the set of 6 measurements.

Rocchio's technique allows us to build a model using the positive data only (in which case no DFO is applied), which leads to $AvgP = 0.657 \pm 0.076$. We will call this the reference value.

6.1. Baseline results

By using the baseline assumption, for each of the models considered (i.e., Rocchio, naive Bayes and the SVM), all unlabeled documents were treated as members of the negative class. The performance obtained with this setup is presented in the first column of results in Tables 3–5, according to which the baseline assumption does actually lead to an above-random performance, suggesting that distribution of the positive documents was significantly different from the distribution of the "world" sample represented by the unlabeled documents. From a practical perspective, this means that an unobtrusive passive-feedback interface, such as the one proposed here, could provide valuable user benefits in a

Table 3. *AvgP* results for **Rocchio/DFO** averaged across the 6-element user set. The *baseline* column corresponds to the mean and standard deviation *AvgP* for baseline assumption, with the remaining columns showing the percentage increase/decrease of mean *AvgP* with reference to the baseline after applying the EM. Different modes of initializing the negative class prior to EM are indicated.

Ratio	Baseline	EM w/const-init	EM w/proximity-init	EM w/tail-free-init	EM w/tail-clamp-init
		Incorporating both positive and negative label estimates			
1/8	0.543 ± 0.034	+20.6	+20.8	+24.3	0
1/4	0.604 ± 0.071	+14.0	+14.3	+10.4	0
1/2	0.697 ± 0.091	−0.4	−0.1	−4.0	0
1	0.721 ± 0.114	−2.9	−2.9	−2.9	−5.2
2	0.670 ± 0.083	−1.8	−1.9	−1.8	−3.3
4	0.651 ± 0.076	−0.9	−0.7	−0.9	−1.2
8	0.643 ± 0.081	−0.8	−0.8	−0.8	0
		Incorporating negative label estimates only			
1/8	0.543 ± 0.034	+18.9	+19.8	+27.2	0
1/4	0.604 ± 0.071	+15.6	+15.6	+17.0	0
1/2	0.697 ± 0.091	+4.2	+4.2	+2.0	0
1	0.721 ± 0.114	−0.3	−0.3	−0.3	−8.2
2	0.670 ± 0.083	+0.3	+0.2	+0.3	−3.3
4	0.651 ± 0.076	+0.2	+0.3	+0.2	−1.2
8	0.643 ± 0.081	−0.5	−0.5	−0.5	0

EM columns show % increase/decrease of mean *AvgP* with respect to the baseline.

Table 4. *AvgP* results for **naive Bayes** averaged across the 6-element user set. The *baseline* column corresponds to the mean and standard deviation *AvgP* for baseline assumption, with the remaining columns showing the percentage increase/decrease of mean *AvgP* with reference to the baseline after applying the EM. Different modes of initializing the negative class prior to EM are indicated.

Ratio	Baseline	EM w/const-init	EM w/proximity-init	EM w/tail-free-init	EM w/tail-clamp-init
		Incorporating both positive and negative label estimates			
1/8	0.650 ± 0.098	+5.2	+5.4	+5.2	0
1/4	0.677 ± 0.120	+5.6	+5.5	+5.6	0
1/2	0.703 ± 0.119	+5.7	+5.6	+5.7	0
1	0.722 ± 0.153	+9.0	+9.0	+9.0	+8.6
2	0.684 ± 0.104	+11.6	+7.9	+10.2	+10.2
4	0.677 ± 0.117	+11.1	+10.9	+6.4	+5.8
8	0.684 ± 0.098	+5.1	+4.4	+4.5	+5.0
		Incorporating negative label estimates only			
1/8	0.650 ± 0.098	+5.5	+5.5	+5.5	0
1/4	0.677 ± 0.120	+5.2	+4.9	+5.2	0
1/2	0.703 ± 0.119	+6.5	+6.4	+6.5	0
1	0.722 ± 0.153	+10.3	+10.3	+10.3	+8.6
2	0.684 ± 0.104	+14.8	+14.8	+12.7	+12.7
4	0.677 ± 0.117	+14.0	+14.3	+13.0	+12.0
8	0.684 ± 0.098	+1.2	+1.5	+7.3	+8.2

EM columns show % increase/decrease of mean *AvgP* with respect to the baseline.

"real-world" system. Interestingly, the performance peaks for the case when the number of unlabeled documents equals the number of positive labeled documents.

6.2. EM results

For each unlabeled/labeled training-data ratio, the EM algorithm was run till convergence (usually 3–7 iterations). EM was implemented in its basic form and used in conjunction with the multinomial naive Bayes network (Nigam et al. 2000). The computation time added by the EM was rarely more than double the time necessary to build and run the naive Bayes model, so for some systems it might be feasible to incorporate such a process on-line. Four variants of negative-class initialization were considered as discussed in Section 4.3.

The result of running EM was a set of relevance probability estimates, $P(R \mid d)$, for elements of the unlabeled set. These values were then incorporated directly into the BIM naive Bayes model (see Eqs. (3–5)). In the case of Rocchio/DFO and the SVM, a hard-threshold decision was made to classify each element, based on the value of $P(R \mid d)$,

Table 5. *AvgP* results for **SVM** averaged across the 6-element user set. The *baseline* column corresponds to the mean and standard deviation *AvgP* for baseline assumption, with the remaining columns showing the percentage increase/decrease of mean *AvgP* with reference to the baseline after applying the EM. Different modes of initializing the negative class prior to EM are indicated.

Ratio	Baseline	EM w/const-init	EM w/proximity-init	EM w/tail-free-init	EM w/tail-clamp-init
		Incorporating both positive and negative label estimates			
1/8	0.680 ± 0.109	+2.1	+2.4	+2.1	0
1/4	0.697 ± 0.101	+0.3	+0.4	+0.3	0
1/2	0.700 ± 0.116	+3.3	+2.7	+3.1	0
1	0.728 ± 0.156	+4.0	+4.1	+4.0	+1.7
2	0.700 ± 0.127	+4.9	+5.0	0	−7.4
4	0.707 ± 0.172	+5.9	+5.4	+2.1	−7.2
8	0.736 ± 0.117	−4.1	−4.1	−8.0	−8.0
		Incorporating negative label estimates only			
1/8	0.680 ± 0.109	+2.1	+2.4	+2.1	0
1/4	0.697 ± 0.101	+2.0	+1.6	+2.0	0
1/2	0.700 ± 0.116	+4.7	+4.3	+4.7	0
1	0.728 ± 0.156	+5.8	+5.8	+5.8	+1.7
2	0.700 ± 0.127	+8.4	+8.6	+5.0	−4.7
4	0.707 ± 0.172	+8.1	+8.1	+7.5	+1.1
8	0.736 ± 0.117	+2.6	+2.5	+2.0	+3.4

EM columns show % increase/decrease of mean *AvgP* with respect to the baseline.

as either relevant or non-relevant. As discussed in Section 4.3, there is an ambiguity regarding the use of the labeling results, where the question is whether such use should be limited to the portion labeled as negative (i.e., $1 - P(R \mid d) > P(R \mid d)$) or whether all documents should be used. Therefore we considered two cases corresponding to both scenarios.

Tables 3–5 illustrate the average precision performance of Rocchio/DFO, naive Bayes and SVM, after the EM assignment of class-membership probabilities to the unlabeled data. For better clarity, columns corresponding to post-EM results show only their relative difference from the baseline. The baseline results are clearly exceeded in virtually all cases, except for *ratio* < 1 with `tail-clamp-init` where post-EM experiments are equivalent to the baseline experiments (i.e., where all unlabeled data were treated as negative examples during training). Interestingly, the `tail-clamp-init` initialization variant performs often poorer than others. This indicates that proximity-based ordering of unlabeled documents used with this option has rather limited accuracy, and that we are better off by letting EM adjust the class-probability estimates for all elements of the unlabeled set. Application of initialization modes other than `const-init` makes little difference (i.e., the process converges to roughly equivalent solutions) suggesting that the baseline assumption provides

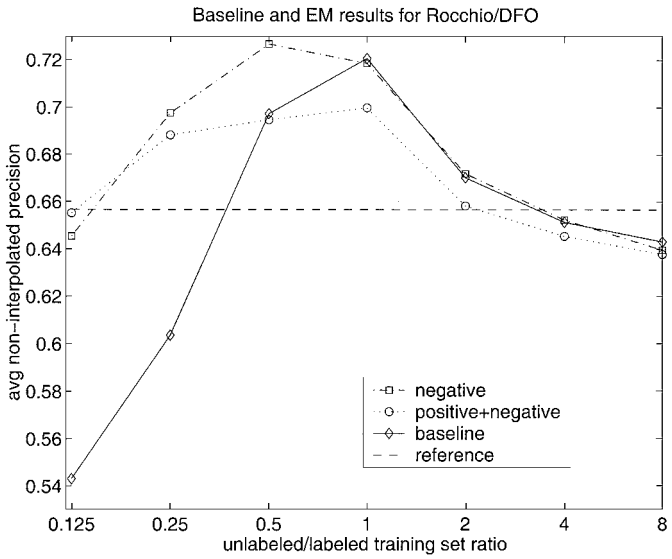

Figure 1. Comparison of **Rocchio/DFO** mean *AvgP* test-set performance for the *baseline* setup as well as two variants of EM labeling utilization, i.e., using the complete labeling output vs. using the negative label estimates only. The horizontal *reference* line corresponds to performance with positive data only (i.e., no unlabeled/negative samples).

a reasonable starting point for the EM. In the following, `const-init` will be used as the basis for comparing different models.

Figures 1–3 show the effectiveness of incorporating EM labeling into the respective algorithms, with and without the use of the positively labeled background documents. The horizontal "reference" line in the figures corresponds to Rocchio trained on positive examples only, which has been depicted for "calibration" purposes. It appears that in all cases there is a disadvantage in incorporating the complete EM output and that the models work best if augmented only with the negatively labeled portion of the background data. This indicates that documents labeled as a result of EM as positive tend to be either redundant with respect to the existing positive training data or, more likely, they represent erroneous labeling, thus introducing undesirable noise into the positive class.

Similarly to the baseline results, best post-EM performance levels are achieved when the number of unlabeled documents approximately matches the size of the positive set which can be understood as follows: For larger bodies of unlabeled documents there is an increased chance that many "true" positive ones will be labeled as negative, which can adversely affect the performance. This is due to the inaccuracy of the baseline assumption used either on its own or to initialize the EM process. In the exceptional case of the SVM, the baseline performance for $ratio = 8$ actually exceeds the performance for $ratio = 1$, which we do not fully understand. For smaller sizes of the unlabeled set, the amount of information to differentiate between positive and negative data is reduced, which makes the models less effective.

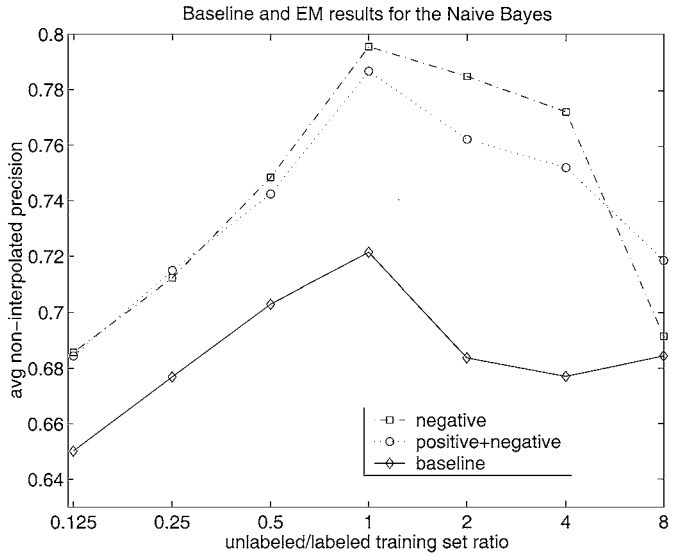

Figure 2. Comparison of **BIM naive Bayes** mean $AvgP$ test-set performance for the *baseline* setup as well as two variants of EM labeling utilization, i.e., using the complete labeling output vs. using the negative label estimates only.

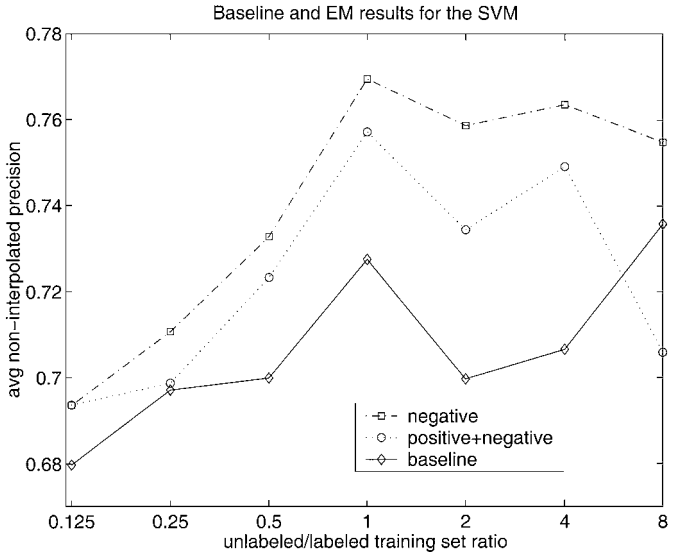

Figure 3. Comparison of **SVM** mean $AvgP$ test-set performance for the *baseline* setup as well as two variants of EM labeling utilization, i.e., using the complete labeling output vs. using the negative label estimates only.

An interesting case is that of Rocchio/DFO where, as shown in figure 1, the performance of the model with positive data only (DFO was not applied in this case) actually exceeds its performance after incorporation of negative data for very large (in the case of baseline also for very small) values of *ratio*. Although we can understand this behavior for large values of *ratio*, the dramatic drop in performance for the baseline variant of Rocchio with small values of *ratio* needs to be explained. We believe that this is caused by the particular way of selecting γ (2) chosen for our experiments. For *ratio* < 1, formula (2) assigns progressively higher weights to members of the negative set, which can result in an elimination of certain important terms characteristic of the positive class (recall that in Rocchio's algorithm only positive-valued weights are allowed so, if an effective negative weight in (1) exceeds its positive counterpart, the term is subsequently ignored). Application of EM helps by identifying the likely positive documents in the unlabeled set and either incorporating them into the positive class or ignoring them altogether.

Figure 4 compares the post-EM performance of Rocchio/DFO, naive Bayes and SVM where only the negative labeling results were used. All algorithms show a similar dependence on the unlabeled/labeled data ratio and, within the best-performance range, naive Bayes demonstrates the highest precision. The performance of SVM is slightly worse although this model is more resilient to extreme values of *ratio*. The lower precision of SVM might be due to its high dependence on the position of the negative-positive class boundary as defined by the training data, which is prone to inaccuracies in the current scenario. On the other hand, when one of the classes has markedly more examples than the other, the

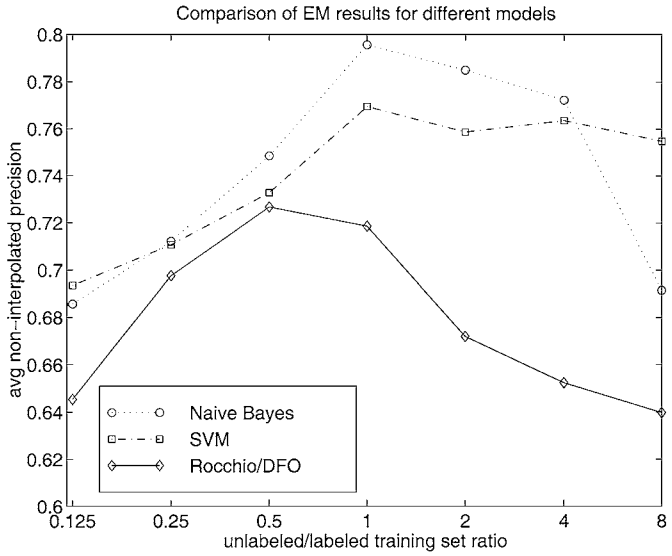

Figure 4. Comparison of average precision test-set performance for the models considered. Post-EM <const-init> results are shown, where *only* the negatively label estimates are utilized. All methods show a similar dependence on the labeled/unlabeled data ratio. *Naive Bayes* is characterized by highest performance, but *SVM* also performs reasonably well.

under-represented class stabilizes the number of support vectors and, therefore, the performance of the SVM does not drop very sharply for extreme values of *ratio*. In our experiments, Rocchio/DFO demonstrated the lowest levels of performance. This is probably caused by the inability of this method to fully take advantage of the information offered by the negative data because terms that are present only in the negative set are ignored. It is possible that a modification of the basic Rocchio algorithm could alleviate this problem.

7. Results (Reuters experiment)

All results presented below refer to averaging across the 100-element synthetic subject set.

7.1. Baseline results

A comparison of the baseline (i.e., all unlabeled documents treated as negative) *AvgP* performance for Rocchio/DFO, naive Bayes and SVM are shown in figure 5. The horizontal

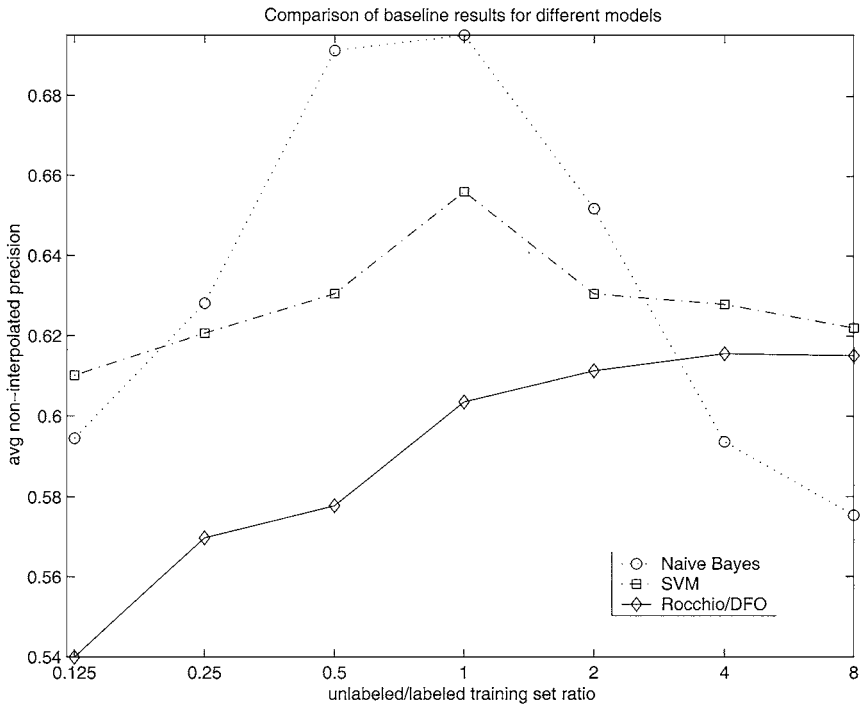

Figure 5. Comparison of average precision test-set performance for the models considered using synthetic user data. Baseline results are shown. *Naive Bayes* is characterized by highest performance, while *SVM* demonstrates the least sensitivity to the amount of unlabeled data used. Unlike the other two models, Rocchio/DFO steadily improves with the amount of unlabeled data used.

reference line corresponds to performance of the Rocchio method using the positive training data only. When compared with the results obtained in the newspaper experiment, it appears that the overall performance level for all methods is lower. This might be due to a better separation of the positive and unlabeled datasets for real users, and also by the fact that, in the case of real users, the data consisted of the leads of articles, which can be considered more focused than the complete article bodies used in the Reuters experiment.

7.2. EM results

Based on the relative insensitivity of EM to different initialization conditions demonstrated in the newspaper experiment, only the `const-init` initialization mode was considered for the Reuters experiment. The proportion of the unlabeled set assigned to the negative class as a result of EM in the newspaper and Reuters experiments is compared in Table 6. As can be seen, with the number of unlabeled documents increasing, their proportion assigned to the positive class also slowly increases, which is consistent for both experiments.

Table 6. Average percentage of unlabeled documents assigned to the negative class as a result of EM (initialized with the `const-init` mode) for the *newspaper* and *Reuters* experiments. In both cases, the percentage gradually decreases as more unlabeled documents are used. The first column designates the unlabeled/labeled document count ratio.

Ratio	Newspaper	Reuters
1/8	80	78
1/4	77	79
1/2	74	77
1	71	79
2	65	73
4	61	69
8	68	61

Table 7. Uniform and weighted mislabeling error rate resulting from the baseline assumption of treating all unlabeled documents as negative.

Ratio	Mislabeling error	Weighted mislabeling error
1/8	0.473	0.663
1/4	0.453	0.649
1/2	0.400	0.596
1	0.372	0.573
2	0.363	0.565
4	0.356	0.557
8	0.345	0.545

Unlike in the previous experiment, however, in this case it was possible to asses the labeling performance due to EM. Based on the category membership of an unlabeled document, its relevance to a particular synthetic user was known. Let us define an error rate for the unlabeled set for a user as the ratio of the number of relevant documents contained in that set to the total number of elements in that set. It quantifies the error being made by assuming all unlabeled documents to be non-relevant. Average values of thus defined error rate for different values of unlabeled/labeled document count ratio are shown in the second column of Table 7. The third column shows weighted error values, which penalize the error of labeling a positive document as negative proportionally to the relevance mixing weight assigned to the document's category while generating a user's profile. As can be expected, the weighted error attains higher values. It can be seen that, in the case of synthetic users, our baseline assumption about the non-relevance of unlabeled documents is largely correct.

The EM process results in assigning a document-relevance probability estimate to each unlabeled document. These estimates can then be applied to label each document as either positive or negative. As discussed in the newspaper experiment, we considered two ways of accommodating these results—one where both the positive and negative labeling outcomes were considered, and one where only the negative labeling results were taken into account. The resulting mislabeling errors are compared in figures 6 and 7. By analyzing the raw

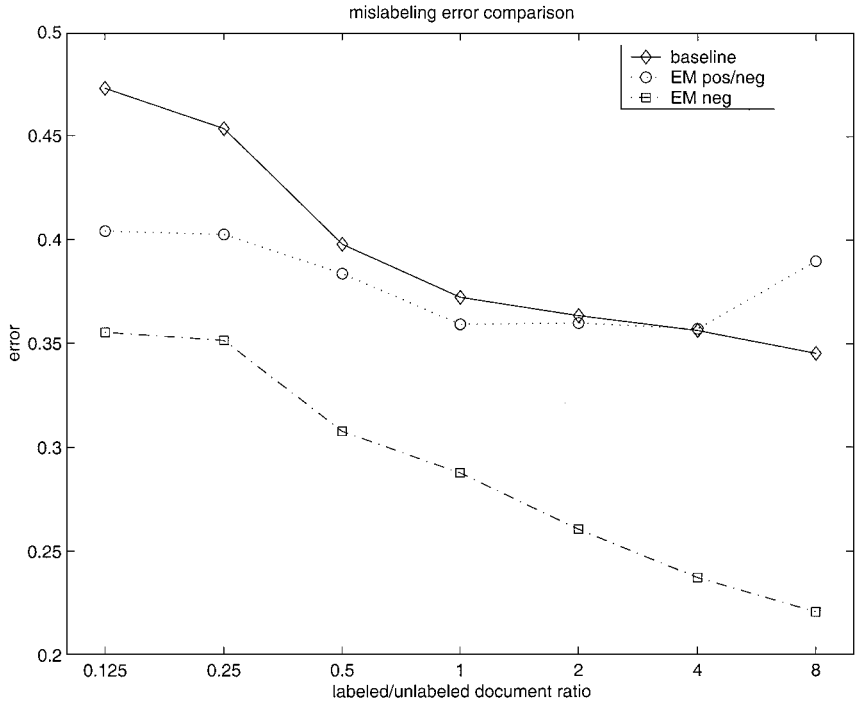

Figure 6. Dependence of the raw (i.e., unweighted) labeling error on the labeled/unlabeled document ratio before and after EM. The two approaches to utlizing EM results are considered (see text for details).

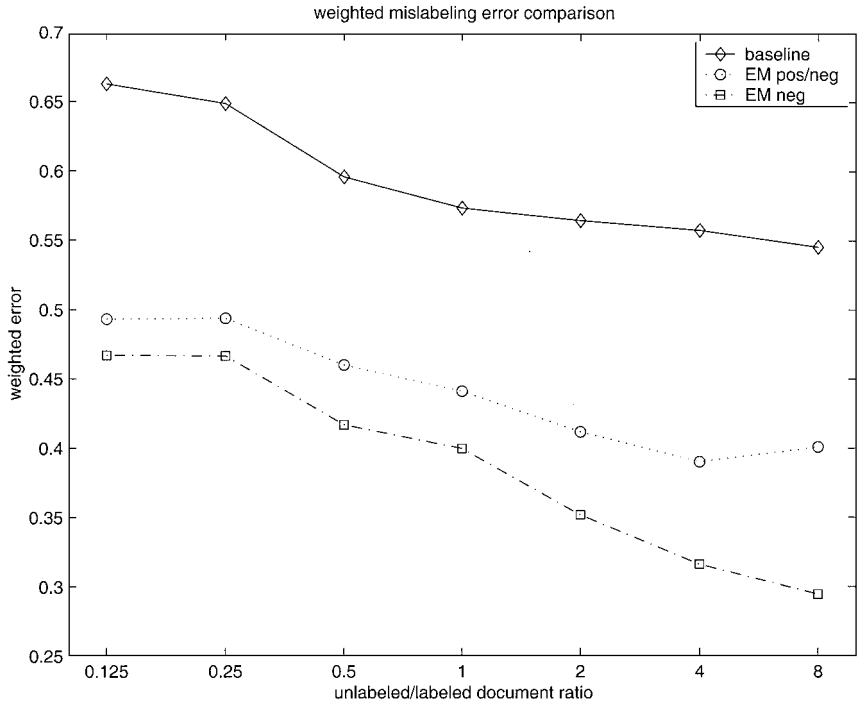

Figure 7. Dependence of the weighted labeling error on the labeled/unlabeled document ratio before and after EM. The two approaches to utlizing EM results are considered (see text for details).

error rate in figure 6, it is apparent that little is gained by incorporating both the positive and negative EM labels. However, figure 7 indicates that EM is indeed able to correctly identify the most relevant documents, even if it mislabels the less relevant ones as non-relevant. This is most likely due to the fact that documents from the most relevant categories have a higher presence in the positive training set. Both the raw and weighted error curves show that EM is quite successful in correctly identifying the true labels of non-relevant documents.

The *AvgP* results for baseline and EM-augmented data for the three algorithms considered are shown in figures 8–10. The dependence of *AvgP* on the unlabeled/labeled document count ratio appears to be more dependent on the particular approach taken, than in the case of the newspaper experiment (see the corresponding figures 1–3). Both naive Bayes and SVM show a performance peak when the document ratio is close to one (similarly as in the newspaper experiment), while Rocchio/DFO appears to be steadily improving when the number of unlabeled documents increases. At the same time, the first two methods achieve significantly higher performance levels than Rocchio.

In the case of Rocchio, the disparity between the newspaper and Reuters experiments is most likely due to different document statistics in the two experiments. In the newspaper experiment, all documents were represented by very short leads and contained words designed to capture the essence of the corresponding full article. In the Reuters experiment, on the other hand, complete documents were used. In the Rocchio algorithm, if the negative weight

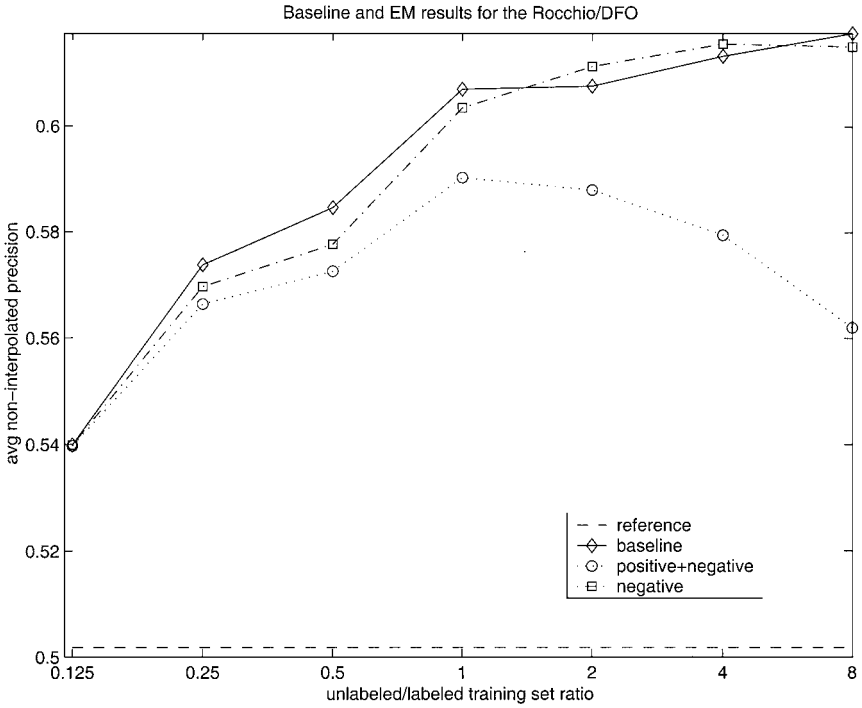

Figure 8. Comparison of **Rocchio/DFO** mean *AvgP* test-set performance for the *baseline* setup as well as two variants of EM labeling utilization, i.e., using the complete labeling output vs. using the negative label estimates only. The horizontal *reference* line corresponds to performance with positive data only (i.e., no unlabeled/negative samples). Unlike in the newspaper experiment, for the simulated data, there is essentially no advantage due to the use of EM.

of a term exceeds its positive counterpart, the term becomes effectively "switched off". Thus, knowing that the majority of the unlabeled document corpus can in fact be considered negative, as the number of unlabeled documents increases, more and more irrelevant and "noisy" terms become inactive, to which we can attribute the apparent improvement in Rocchio's performance. It can be expected however that, for larger amounts of negative data, some relevant terms will also become deactivated and lead to a gradual drop in Rocchio accuracy. In the case where both the positive and negative post-EM labeling results are used, the errors committed when labeling a negative document as positive appear to have detrimental effect and, as the number of such contributions increases relative to the size of the positive training set, the overall performance degrades. Interestingly, the post-EM performance of naive Bayes seems to be very close to the baseline (especially for smaller amounts of unlabeled data), while SVM appears to benefit from the EM to a much larger extent.

Figure 11 compares the post-EM performance of Rocchio/DFO, naive Bayes and SVM where only the negative labeling results were used. As in the newspaper experiment, within the best-performance range naive Bayes demonstrates the highest precision, while SVM is much more resilient to extreme values of the unlabeled/labeled document count ratio.

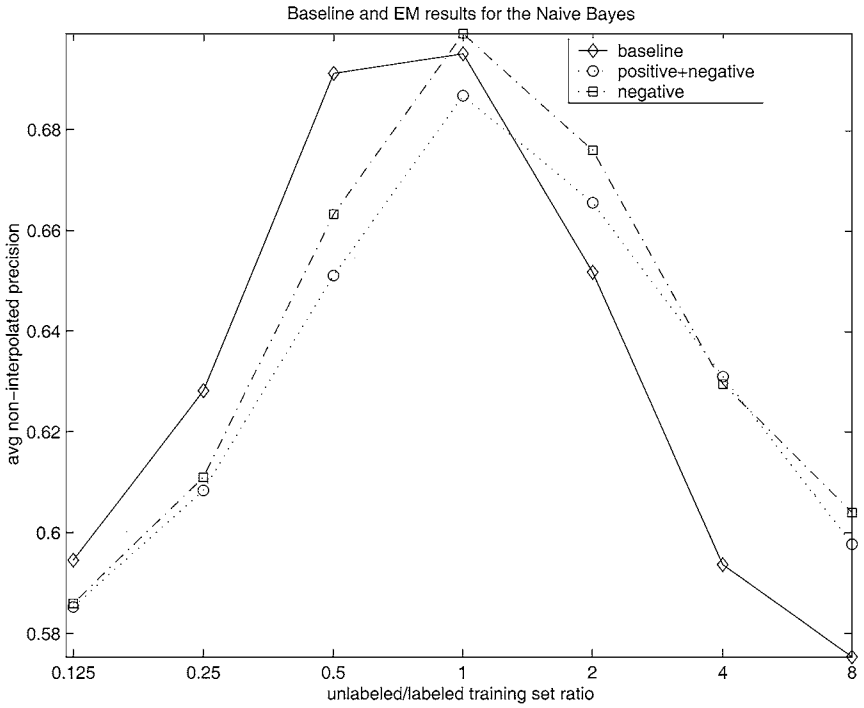

Figure 9. Comparison of **BIM naive Bayes** mean *AvgP* test-set performance for the *baseline* setup as well as two variants of EM labeling utilization, i.e., using the complete labeling output vs. using the negative label estimates only. The benefit of using the EM becomes apparent only for larger quantities of unlabeled data.

The dependence of Rocchio on the value of *ratio* is qualitatively different (as discussed above) from the one showed in the newspaper experiment, although it proved to be the worst performer as before.

8. Conclusions

We addressed the problem of compensating for the lack of definite negative labeled data in the user preference ranking related to personalizing on-line news. Since the data obtained from real users were relatively scarce, to verify our results on a larger dataset, a set of 100 synthetic users was created, whose interests were modeled as mixtures of topics from the Reuters dataset. Although the document statistics of the real and simulated users were quite different, the qualitative results obtained in both cases proved to be very close. Interestingly, the benefit of applying the EM procedure was higher in the case of real-user data, suggesting the existence of a better-defined separation between relevant and non-relevant documents in that case.

The ranking scenario can be considered as a special case of a binary classification problem where data items (documents/articles) have certain probabilities of belonging to the positive (relevant) or negative (non-relevant) class. Ideally, a training set of labeled positive/negative

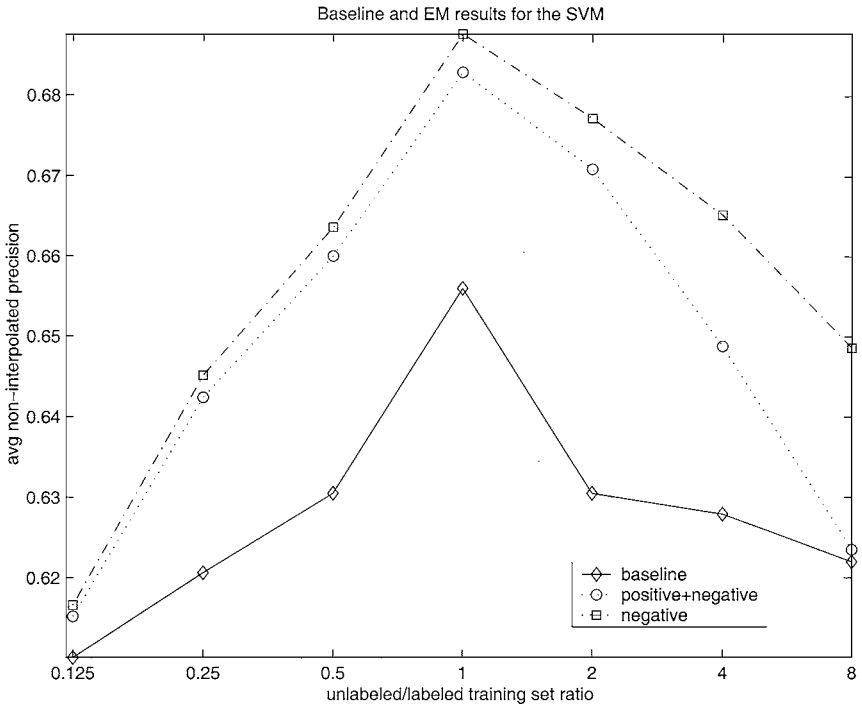

Figure 10. Comparison of **SVM** mean *AvgP* test-set performance for the *baseline* setup as well as two variants of EM labeling utilization, i.e., using the complete labeling output vs. using the negative label estimates only. The SVM model clearly benefits from the application of EM.

documents should be used to create a user model guaranteeing optimum ranking, where the most relevant documents receive the highest rank, but the problem is compounded by a complete lack of the negative labeled examples. Fortunately, if an unlabeled corpus of data is available, it can be used effectively to estimate the negative class and thus enhance the performance of various preference models. In particular, we based our experimental work on three popular techniques: the Rocchio's algorithm with Dynamic Feedback Optimization (DFO), the naive Bayes classifier and Support Vector Machines (SVM).

We investigated the validity of a simple baseline assumption which treats all unlabeled data as negative. The baseline assumption about non-relevance of the background documents appears to be remarkably correct. Although we have no doubt that active feedback approaches could lead to a significant performance increase of a ranking system (McCallum and Nigam 1997), it seems that fully automatic on-line systems based on passive feedback also have merit and could be quite attractive in practical applications. Their value is increased by their unobtrusiveness, since no explicit rating is required of the users.

Although the baseline assumption can lead to reasonable performance levels, the utility of the unlabeled data can be significantly increased by means of the EM algorithm, which was implemented in conjunction with the multinomial naive Bayes classifier. Importantly, the EM/naive Bayes combination proved to be fairly insensitive to different ways of

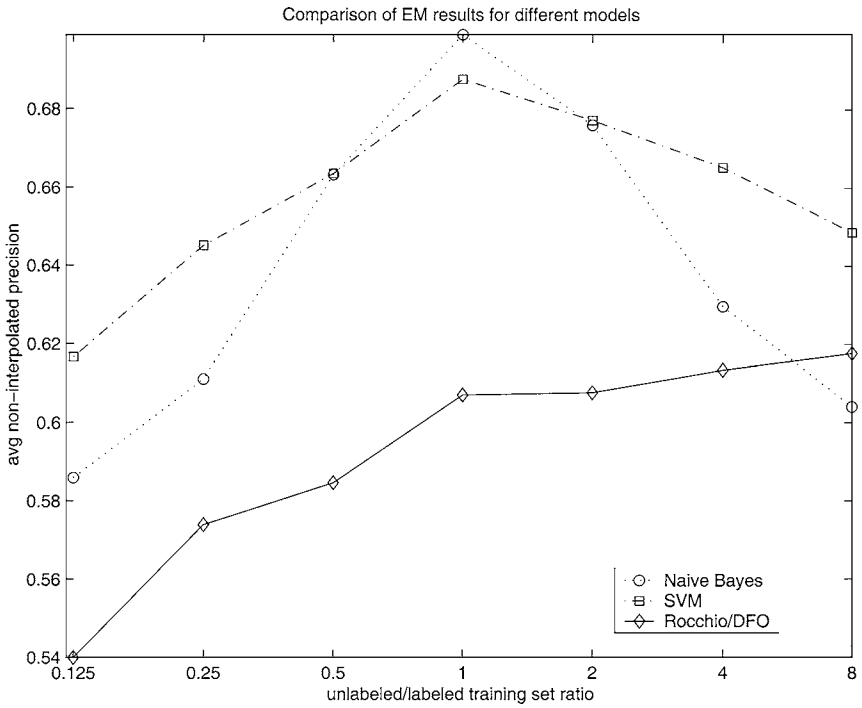

Figure 11. Comparison of average precision test-set performance for the models considered using synthetic user data. Post-EM <const-init> results are shown, where *only* the negatively label estimates are utilized. *Naive Bayes* is characterized by highest performance but *SVM*, while also performing reasonably well, shows much smaller sensitivity to the amount of unlabeled data used. Unlike the other two models, Rocchio/DFO steadily improves with the amount of unlabeled data used.

initializing the negative class and the baseline assumption (i.e., initially treating all unlabeled documents as members of the negative class) can be used effectively for that purpose. For all of the models considered in this work, post-EM labeling led to an increased ranking accuracy, especially when the number of unlabeled documents was approximately equal to the number of the labeled positive documents. Interestingly, when the number of unlabeled documents diverges (i.e., grows or shrinks) with respect to the labeled positive set, the post-EM performance seems to degenerate and approach that of the baseline. We believe that when the number of unlabeled documents grows this behavior is caused by the inherent inaccuracy in initializing the EM algorithm due to the lack of a definite negative set. With increasing amounts of unlabeled data, the mistakes in estimating their labels tend to outweigh the influence of the initial labeled data set. On the other hand, when the number of unlabeled documents is small with respect to the positive data, the models have a diminished ability of discriminating between the two classes and hence the drop in performance.

Of the models considered here, naive Bayes proved to be most accurate, with SVM providing a comparable level of performance. SVM benefited the most from filtering of the unlabeled training data due to EM, since the model is inherently sensitive to the similarities

between the positive and negative training examples, and EM proved successful in increasing the proportion of the negative data in the unlabeled set.

One of the questions we posed was whether we should apply EM to obtain just the negatively labeled training data or should we also use the positive label estimates. The experimental results seem to indicate that adding the positive label estimates actually harms the ranking performance when compared to the case when only the negative label estimates are used. This is likely due to violation of the assumptions made by EM regarding the properties of the input data. Namely, for each user, both the relevant and non-relevant document documents most probably have a multi-cluster structure (while the EM is trying to model them with just one mixture component per the relevant class and one mixture component per the non-relevant class), and the relevant/non-relevant classification boundary does not necessarily have to separate the natural clusters recoverable from the input data. Our results appear thus to support the observations made in (Nigam et al. 2000), where the authors proposed certain extensions to the general methodology which can overcome the problems described above. It was suggested to modify the EM procedure such that the contribution of the unlabeled data to the parameter-estimation process is smaller than that of the labeled data and also to associate each class label with several—not just one—mixture components. We would like to consider these extensions in future work.

In most cases, the algorithms we considered proved to be sensitive to the balance between positive and negative examples used (with the latter ones being derived from the unlabeled data). This largely conforms with the past research on using sampling techniques to "equalize" class membership during training. Although certain techniques of overcoming the basic class-imbalance problems are known, it has to be noted that the methods used in this work were applied in their standard form (without attempting to implement asymmetric misclassification loss functions, for example). The sampling of unlabeled data to be used in training was also carried out in a blind (i.e., uniform) fashion. Recently, it has been observed that guided sampling (i.e., trying to preserve the mixture distribution of the sampled data) might be more beneficial in imbalanced problems (Nickerson et al. 2001). In future research, we intend to use such, and related, techniques to better utilize the volume of unlabeled data available.

Notes

1. http://www.newshound.com
2. http://www.crayon.com
3. http://www.pointcast.com
4. http://ais.gmd.de/~thorsten/svm_light/
5. http://www.research.att.com/~lewis/reuters21578.html

References

Billsus D and Pazzani M (1999) A hybrid user model for news story classification. In: Seventh International Conference on User Modeling (UM '99). http://www.ics.uci.edu/~pazzani/Publications/um99.ps.

Blum A and Mitchell T (1998) Combining labeled and unlabeled data with co-training. In: Proceedings of the 1998 Conference on Computational Learning Theory.

Buckley C and Salton G (1995) Optimization of relevance feedback weights. In: Proceedings of 18th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 351–357.

Buckley C, Salton G, Allan J and Singhal A (1995) Automatic query expansion using SMART: TREC-3. In: Harman DK, Ed., Proceedings of the 3rd Text Retrieval Conference (TREC-3) (NIST SP 500-225).

Claypool M, Gokhale A, Miranda T, Murnikov P, Netes D and Sartin M (1999) Combining content-based and collaborative filters in an online newspaper. ACM SIGIR Recommender Systems Workshop. http://www.cs.wpi.edu/∼claypool/papers/content-collab/content-collab.ps.

Cohn D, Atlas L and Ladner R (1994) Improving generalization with active learning. Machine Learning, 15(2):201–221.

Croft W and Harper D (1979) Using probabilistic models of document retrieval without relevance information. Journal of Documentation, 35:285–295.

Dempster AP, Laird NM and Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B, 39:1–38.

Domingos P (1999) MetaCost: A general method for making classifiers cost-sensitive. In: Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining (KDD-99).

Foltz PW and Dumais ST (1992) Personalized information delivery: An analysis of information filtering methods. Communications of the ACM, 35(12):51–60. http://www-psych.nmsu.edu/∼pfoltz/cacm/cacm.html.

Frakes WB and Baeza-Yates R (1992) Eds., Information Retrieval: Data Structures and Algorithms. Prentice-Hall, Boston, MA.

Harman D (1992) Ranking algorithms. In: Frakes WB and Baeza-Yates R, Eds., Information Retrieval: Data Structures and Algorithms. Prentice-Hall, Boston, MA, pp. 363–392.

Iyengar V, Apte C and Zhang T (2000) Active learning using adaptive resampling. In: Proceedings of ACM SIGKDD 2000.

Japkowicz N (2000) The class imbalance problem: Significance and strategies. In: Proceedings of the 2000 International Conference on Artificial Intelligence. http://www.cs.dal.ca/∼nat/Papers/ic-ai-2000.ps.

Jennings A, Higichi H and Liu H (1993) A user model neural network for a personal news service. Australian Telecommunication Research, 27(1):1–12.

Joachims T (1997) Text categorization with support vector machines: Learning with many relevant features. Technical Report LS-8/23, University of Dortmund.

Joachims T (1999) Making large-scale svm learning practical. In: Schoelkopf B, Burges C and Smola A, Eds., Advances in Kernel Methods—Support Vector Learning. MIT Press.

Joachims T, Freitag D and Mitchell T (1997) Webwatcher: A tour guide for the world wide web. In: Proceedings of the International Joint Conference on Artificial Intelligence. http://www.cs.cmu.edu/afs/cs/project/theo-6/web-agent/www/ijcai97.ps.

Kamba T, Sahagami H and Koseki Y (1997) ANATANAGONOMY: A personalized newspaper on the world wide web. Int. J. Human-Compuer Studies, 46:789–803.

Kubat M, Holte R and Matwin S (1997) Learning when negative examples abound. In: Proceedings of the European Conference on Machine Learning, ECML'97, pp. 146–153. http://www.cacs.louisiana.edu/∼mkubat/publications/imbalanced.ps.

Kubat M and Matwin S (1997) Addressing the curse of imbalanced training sets: One-sided selection. In: Proceedings of the 14th International Conference on Machine Learning, ICML'97, pp. 179–186. http://www.cacs.louisiana.edu/∼mkubat/publications/sampling.ps.

Lang K (1995) NewsWeeder: Learning to filter NetNews. In: Proceedings of the 12th International Conference on Machine Learning: ICML-95, pp. 331–339.

Lewis DD (1998) Naive (bayes) at forty: The independence assumption in information retrieval. In: Proceedings of the 10th European Conference on Machine Learning, pp. 4–15. http://www.research.att.com/∼lewis/papers/lewis98b.ps.

Lewis DD and Gale WA (1994) A sequential algorithm for training text classifiers. In: Croft WB and van Rijsbergen CJ, Eds., SIGIR 94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Springer-Verlag, Dublin, Ireland, pp. 3–12.

Lieberman H (1995) Letizia: An agent that assists web browsing. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 924–929. http://lieber.www.media.mit.edu/people/lieber/Lieberary/Letizia/Letizia-AAAI/Letizia.html.

Losee RM (1998) Text Retrieval & Filtering: Analytic Models of Performance. Kluwer Academic Publishers, New York.

McCallum AK and Nigam K (1997) Employing EM in pool-based active learning for text classification. In: Proceedings of the 1998 International Machine Learning Conference, pp. 25–32. http://www.cs.cmu.edu/∼mccallum/papers/emactive-icm198.ps.gz.

McCallum AK and Nigam K (1998) A comparison of event models for naive bayes text classification. AAAI-98 Workshop on Learning for Text Categorization. http://www.cs.cmu.edu/∼mccallum/papers/multinomial-aaai98w.ps.

McLachlan GJ and Krishnan T (1996) The EM Algorithm and Extensions. John Wiley & Sons, Philadelphia, PA.

Mitra M, Singhal A and Buckley C (1997) Learning queries in a query zone. In: Proceedings of the 20th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 25–32.

Morik K, Imboff M, Brockhausen P, Joachims T and Gather U (2000) Knowledge discovery and knowledge validation in intensive care. Artificial Intelligence in Medicine, 19(3):225–249.

Morita M and Shinoda Y (1994) Information filtering based on user behavior analysis and best match text retrieval. In: Proceedings of 17th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 272–281.

Nickerson A, Japkowicz N and Milios E (2001) Using unsupervised learning to guide resampling in imbalanced data sets. In: Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics.

Nigam K and Ghani R (2000) Analyzing the effectiveness and applicability of co-training. In: Proceedings of the Ninth International Conference on Information and Knowledge Management.

Nigam K, McCallum AK, Thrun S and Mitchell T (2000) Text classification from labeled and unlabeled documents using EM. Machine Learning, 39(2):103–134.

Pazzani M and Billsus D (1997) Learning and revising user profiles: The identification of interesting web sites. Machine Learning, 27:313–331. http://www.ics.uci.edu/∼pazzani/publications/SW-MLJ.pdf.

Pazzani M, Merz C, Murphy, P, Ali K, Hume T and Brunk C (1994) Reducing misclassification costs. In: 11th International Conference of Machine Learning, pp. 217–225. http://www.ics.uci.edu/∼pazzani/publications/MLC94.pdf.

Pednault E, Rosen BK and Apte C (2000) Handling imbalanced data sets in insurance risk modeling. Technical Report RC-21731, IBM.

Platt J (2000) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Smola A, Bartlett P, Scholkopf B and Schuurmans D, Eds., Advances in Large-Margin Classifiers (Neural Information Processing), MIT Press,

Porter M (1980) An algorithm for suffix stripping. Program (Automated Library and Information Systems), 14(3):130–137.

Provost F (2000) Machine learning from imbalanced data sets 101. In: Japkowicz N, Ed., Learning from Imbalanced Data Sets—Papers from the AAAI Workshop. AAAI Press, Austin, TX, pp. 1–3.

Robertson SE and Sparck-Jones K (1976) Relevance weighting of search terms. JASIS pp. 129–176.

Rocchio J (1971) Relevance feedback in information retrieval. In: Salton G, Ed., The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice-Hall, pp. 313–323.

Salton G (1971) The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice-Hall, Melbourne, Australia.

Schapire RE, Singer Y and Singhal, A (1998) Boosting and Rocchio applied to text filtering. In: Proceedings of 21st International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 215–223. http://www.research.att.com/∼schapire/cgi-bin/uncompress-papers/SchapireSiSi98.ps.

Schohn G and Cohn D (2000) Less is more: Active learning with support vector machines. In: Proceedings of the Seventeenth International Conference on Machine Learning.

Seung HS, Opper M and Sompolinsky H (1992) Query by committee. In: Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory, pp. 287–294.

Stevens C (1992) Automating the creation of information filters. Communications of the ACM, 35(12):48.

van Rijsbergen CJ (1979) Information Retrieval. Buttersworth, London.

Vapnik VN (1998) Statistical Learning Theory. John Wiley, New York.

Yang Y and Liu X (1999) A re-examination of text categorization methods. In: Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 42–49. http://www.cs.cmu.edu/∼yiming/papers.yy/sigir99.ps.