Chapter 1

Adaptive Information Filtering

Yi Zhang, University of California Santa Cruz

1.1 Introduction

A financial analyst wants to be alerted of any information that may affect the price of the stock he is tracking; an agent working in the Homeland Security Department wants to be alerted of any information related to potential terror attacks; a customer call center representative wants to answer customer calls about problems that he can handle; and a student wants to be alerted of fellowship or financial aid opportunities appropriate for her/his circumstances.

In these examples, the user preferences are comparatively stable and represent a long term information need, the information source is dynamic, information arrives sequentially over time, and the information needs to be delivered to the user as soon as possible. Traditional ad hoc search engines, which are designed to help the users to pull out information from a comparatively static information source, are inadequate to fulfill the requirements of these tasks. Instead, a filtering system can better serve the user. A filtering system is an autonomous agent that delivers good information to the user in a dynamic environment. As opposed to forming a ranked list, it estimates whether a piece of information matches the user needs as soon as the information arrives and pushes the information to the user if the answer is "yes", so a user can be alerted of any important information on time.

A typical information filtering system is shown in Figure 1.1. In this figure, a piece of information is a document. A user's information needs are

0-8493-0052-5/00/$0.00+$.50

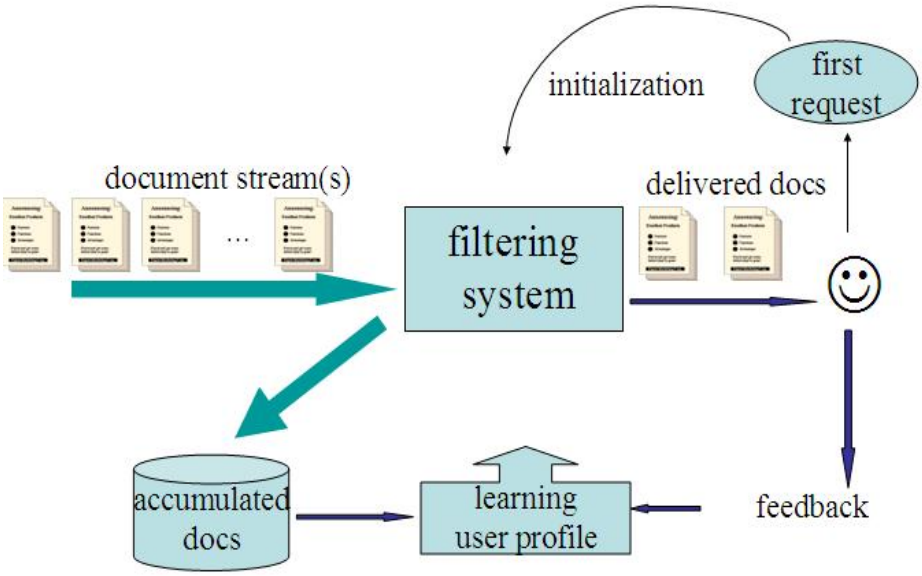

FIGURE 1.1: A typical filtering system. A filtering system can serve many users, although only one user is shown in the figure. Information can be documents, images, or videos. Without lose of generality, we focus on text documents in this chapter.

represented in a user profile. The profile contains one or more classes, such as "stock" or "music", and each class corresponds to one information need. When a user has a new information need, he/she sends to the system an initial request, such as a query or an example of what he/she wants. The system then initializes and creates a new online classifier in the user's profile to serve this information need. As future documents arrive, the system delivers documents the classifier considered relevant to the user. The user may then read the delivered documents and provide explicit feedback, such as identifying a document as "good" or "bad". The user also provides some implicit feedback, such as deleting a document without reading it or saving a document. The filtering system uses the user feedback accumulated over time to update the user profile.

Adaptive filtering vs. retrieval: standard ad hoc retrieval systems, such as search engines, let users use short queries to pull information out of a repository. These systems treat all users the same given the same query. Most IR systems return back documents that match a user query. They assume that a user knows what he/she wants, and what words to use to describe it whenever he/she has an information need. However, a user often does not know these or thinks he/she needs to know one thing but actually needs something else. For example, a financial analyst may search for news in order to check whether the earnings of a company matches the projected earnings. However, also relevant to this task is the large number of customer complaints about the company's product in the blog space. Another example is a research scientist often wants to keep up-to-date with what is happening within a research field, but not looking for a specific answer [12].

If the information need of a user is more or less stable over a long period of time, a filtering system is a good environment to learn user profiles (also called user models) from a fair amount of user feedback that can be accumulated over time. In other words, the adaptive filtering system can serve the user better by learning user profiles while interacting with the user, thus information delivered to the user can be personalized to an individual user's information needs automatically. Even if the user's interest drifts or changes, the adaptive filtering system can still adapt to the user's new interest by constantly updating the user profile from training data, creating new classes automatically [30], or letting the user create/delete classes.

Adaptive filtering vs. collaborative filtering: Collaborative filtering is an alternative approach used by *push* system to provide personalized recommendations to users. Adaptive filtering, which is also called content based filtering, assumes what a user will like is similar to what the user liked before, and thus make recommendations for one user based on the user's feedback about past documents. Collaborative filtering assumes users have similar tastes on some items may also have similar preferences on other items, and thus make recommendations for one user based on the feedback from other users that are similar to this user. Memory-based heuristics and model based approaches have been used in collaborative filtering task [42] [29] [18] [37] [35]

[34] [15] [67]. This chapter does not intend to compare adaptive filtering with collaborative filtering or claim which one is a better. We think each complements the other. Adaptive filtering is extremely useful for handling new documents/items with little or no user feedback, while collaborative filtering leverages information from other users with similar tastes and preferences in the past. Researchers [52] [11] [77], have found that a recommendation system will be more effective when both techniques are combined. However, this is beyond the scope of this chapter and thus not discussed here.

Adaptive filtering vs. Topic Detection and Tracking: The supervised tracking task at the Topic Detection and Tracking (TDT) Workshops is a forum closely related to information filtering [1] [79]. TDT research focuses on discovering topically related material in streams of data. TDT is different from adaptive filtering in several aspects. In TDT, a topic is user independent and defined as an event or activity, along with all directly related events and activities. In adaptive filtering, an information need is user specific and has a broader definition. A user information needs may be a topic about a specific subject, such as "2004 presidential election", or not, such as "weird stories". However, TDT-style topic tracking and TREC-style adaptive filtering have much in common, especially if we treat a topic as a form of user information need. Since a separate chapter in this book is devoted to TDT, we refer the readers to that chapter for research on TDT.

This chapter is organized as follows. Section 1.2 introduces the standard evaluation measures used in the TREC adaptive filtering task. Section 1.3 introduces commonly used retrieval models and adaptive filtering approaches. Section 1.4 describes how to solve the "cold start" problem for new users using Bayesian prior learned from other users. Section 1.5 introduces techniques to avoid redundant information while filtering. This chapter ends with discussion and references to other important topics not covered in details in this book.

1.2 Standard evaluation measures

There are large amounts of prior work on information filtering [31]. The Filtering Track [62] [61] [63] at the Text REtrieval Conference (TREC) is the best known forum for the evaluation of related research [76].

The TREC conference, co-sponsored by the National Institute of Standards and Technology (NIST) and the Defense Advanced Research Projects Agency (DARPA), was started in 1992 and held annually to support research within the information retrieval community by "providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies".

A TREC conference consists of a set of tracks. Each track is an area of

focus in which particular retrieval tasks are defined. The Filtering Track[1] is one of them. In the Filtering Track, the most important research task is the adaptive filtering task, which is designed to model the text filtering process from the moment of profile construction. In this task, the user's information need is stable while the incoming new document stream is dynamic. For each user profile, a random sample of a small amount (2 or 3) of known relevant documents were given to the participating systems, and no relevance judgments for other documents in the training set were available. When a new document arrives, the system needs to decide whether to deliver it to the user or not. If the document is delivered, the user's relevance judgment for it will be released to the system immediately to simulate the scenario that an explicit user feedback is provided to the filtering system by the user. If the document is not delivered, the relevance judgment will never be released to the system. Once the system makes the decision of whether to deliver the document or not, the decision is final [62]. This is strict and not always necessary in a real filtering system, however it is a simple, reasonable and implementable scenario where comparison of the performance between laboratory systems is possible. The goal is to mimic realistic situations where an adaptive filteirng would be used.

In each Filtering Track, NIST provides a test set of documents and relevance judgments. The judgement of relevance is based on topical relevance only. Participants run their own filtering systems on the data, and return to NIST their results. Thus different systems' performance on the set of standard Filtering Track evaluation data sets are publicly available for cross system comparison.

In the information retrieval community, the performance of an ad hoc retrieval system is typically evaluated using relevance-based recall and precision at a certain cut-off of the ranked result. Taking a 20-document cut-off as an example:

$$precision = \frac{the\ number\ of\ relevant\ documents\ among\ the\ top\ 20}{20} \quad (1.1)$$

$$recall = \frac{the\ number\ of\ relevant\ documents\ in\ the\ top\ 20}{all\ relevant\ documents\ in\ the\ corpus} \quad (1.2)$$

What is a good cut off number is unknown. In order to compare different algorithms without a specific cut off, the mean of the precision scores after each relevant document is retrieved, which is called Mean Average Precision (MAP), is often used.

However, the above evaluation measures are not appropriate for filtering. Instead of a ranking list, a filtering system makes an explicit binary decision of whether accept or reject a document for each profile. A *utility function*

[1]The Filtering Track was last run in TREC 2002

FIGURE 1.1: The values assigned to relevant and non-relevant documents that the filtering system did and did not deliver. R^-, R^+, N^+ and N^- correspond to the number of documents that fall into the corresponding category. A_R, A_N, B_R and B_N correspond to the credit/penalty for each element in the category.

	Relevant	*Non-Relevant*
Delivered	R^+, A_R	N^+, A_N
Not Delivered	R^-, B_R	N^-, B_N

is usually used to model user satisfaction and evaluate a system. A general form of the linear utility function used in the recent TREC Filtering Track is shown below [59].

$$U = A_R \cdot R^+ + A_N \cdot N^+ + B_R \cdot R^- + B_N \cdot N^- \tag{1.3}$$

This model corresponds to assigning a positive or negative value to each element in the categories of Table 1.1, where R^-, R^+, N^+ and N^- correspond to the number of documents that fall into the corresponding category, A_R, A_N, B_R and B_N correspond to the credit/penalty for each element in the category. Usually, A_R is positive, and A_N is negative. Assigning a negative weight for a non-relevant document delivered is reasonable considering the fact that retrieving non-relevant documents have a much worse impact than not retrieving relevant ones [13]. B_N and B_R are set to zero in TREC to avoid the dominance of undelivered documents on the final evaluation results, because the number of undelivered documents is usually extremely large and a user's satisfaction is mostly influenced by what the user has seen. The total number of relevant documents $R^+ + R^-$ is a constant for a user profile. Thus a non-zero $A_R \cdot R^+$ already implicitly encodes the influence of R^- undelivered relevant documents in the final evaluation measure.[2]

In the TREC-9, TREC-10 and TREC-11 Filtering Tracks, the following utility function was used:

$$T11U = T10U = T9U = 2R^+ - N^+ \tag{1.4}$$

If we use the T11U utility measure directly and get the final result by averaging across user profiles, profiles with many delivered documents will dominate the final result. So a normalized version T11SU was also used in TREC-11:

$$T11SU = \frac{max(\frac{T11U}{MaxU}, MinNU) - MinNU}{1 - MinNU} \tag{1.5}$$

[2]Some other evaluation measures, such as the normalized utility measure described in the following paragraphs, also consider N^+ implicitly.

where $MaxU = 2*(R^+ + R^-)$ is the maximum possible utility,[3] and $MinNU = -0.5$. If the score is below $MinNU$, the $MinNU$ is used, which simulates the scenario that the users stop using the system when the performance is too poor.[4]

In general, when we average across user profiles to evaluate a filtering system, we can view T11U as micro average measures, while the normalized utility T11SU as macro average measures.

Notice that in a real scenario, we could define user-specific utility functions to model user satisfaction and evaluate filtering systems. A better choice of A_R, A_N, B_R and B_N would depend on the user, the task, and the context. For example, when a user is reading news with a wireless phone, he may have less tolerance for non-relevant documents delivered and prefer higher precision, and thus use a utility function with larger penalty for non-relevant documents delivered, such as $U_{wireless} = R^+ - 3N^+$. When a user is doing research about a certain topic, he may have a high tolerance for non-relevant documents delivered and prefer high recall, and thus use a utility function with less penalty for non-relevant documents delivered, such as $U_{research} = R^+ - 0.5N^+$. When monitoring potential terrorist activities, missing information might be crucial and B_R may be a big non-zero negative value.

In addition to the linear utility measure, other measures such as F-beta [62] defined by van Rijsbergen and DET curves [50], are also used in the research community. Measures that consider novelty or properties of a document have also been proposed by researchers [84].

This section focused on standard evaluation measures used in the TREC adaptive filtering task [62], because they were selected by researchers working in adaptive filtering area over the last several years. Also, the benchmark performance of different systems on several standard filtering data sets are publicly available at http://trec.nist.gov. When a new filtering algorithm is developed to work in a scenario similar to TREC adaptive filtering task, it is often a good practise to evaluate the empirical performance of the new algorithm on the benchmark data sets. As a tradition of the information retrieval community, this becomes a very important method to evaluate the contribution of a new algorithm.

[3] Notice the normalized version does take into consideration undelivered relevant documents. Therefore, it also provides some information about the recall of the system implicitly.

[4] This is not exactly the same, since in TREC the system is evaluated at the very end of filtering process.

1.3 Standard retrieval models and filtering approaches

In this section, we first review some existing information retrieval models since most of them have been adapted, or can be adapted, for the information filtering task. Then we review three common filtering approaches for learning user profiles from explicit user feedback.

We introduce these existing approaches and their drawbacks here, so that the readers can get a better understanding of the common practises in adaptive filtering. This section also provides the context and motivation of the research work described in the following sections. As there is a large amount of literature about standard retrieval models and filtering approaches, we will only review them concisely. For more detail about these models, the readers are referred to other chapters of the book as well as [3] [8] [19] [38] [27] [45] [80] [20] [60] [5] [58] [9] [69] [74] [48] [26] [71] [31] [62] [71] [51] [43].

1.3.1 Existing retrieval models

Information filtering has a long history dating back to the 1970s. It was created as a subfield of the more general information retrieval field, which was originally established to solve the ad hoc retrieval task.[5] For this reason, early work tended to view filtering and retrieval as "two sides of the same coin" [14]. The duality argument is based on the assumptions that documents and queries are interchangeable. This dual view has been questioned [65] [19] by challenging the interchangeability of documents and queries due to their asymmetries of representation, ranking, evaluation, iteration, history and statistics. However, the influence of retrieval models on filtering is still large, because the retrieval models were comparatively well studied and the two tasks share many common issues, such as how to handle words and tokens, how to represent a document, how to represent a user query, how to understand relevance, and how to use relevance feedback. So it is worthwhile to look at various models used in IR and how relevance feedback is used in these models.

In the last several decades, many different retrieval models have been developed to solve the ad hoc retrieval task. In general, there are three major classes of IR models:

[5] Historically, information retrieval was first used to refer to the ad hoc retrieval task, and then was expanded to refer to the broader information seeking scenario that includes filtering, text classification, question answering and more.

1.3.1.1 Boolean models

The *Boolean model* is the simplest retrieval model based on Boolean algebra and set theory. The concept is very simple and intuitive. The drawbacks of the Boolean model are in two aspects: 1) The users may have difficulty to express their information needs using Boolean expressions; and 2) The retrieval system can hardly rank documents since a document is predicted to be either relevant or non relevant without any notion of degree of relevance. Nevertheless, the Boolean model is widely used in commercial search engines because of its simplicity and efficiency. How to use relevance feedback from the user to refine a Boolean query is not straightforward, so the Boolean model was extended for this purposes [47].

1.3.1.2 Vector space models

The *vector model* is a widely implemented IR model, most famously built in the SMART system [69]. It represents documents and user queries in a high dimensional space indexed by "indexing terms", and assumes that the relevance of a document can be measured by the similarity between it and the query in the high dimensional space [68]. In the vector space framework, relevance feedback is used to reformulate a query vector so that it is closer to the relevant documents, or for query expansion so that additional terms from the relevant documents are added to the original query. The most famous algorithm is the Rocchio algorithm [66], which represents a user query using a linear combination of the original query vector, the relevant documents centroid, and the non-relevant documents centroid.

A major criticism for the vector space model is that its performance depends highly on the representation, while the choice of representation is heuristic because the vector space model itself does not provide a theoretical framework on how to select key terms and how to set weights of terms.

1.3.1.3 Probabilistic models

Probabilistic models, such as the *Binary Independence Model (BIM)* ([60]), provide direct guidance on term weighting and term selection based on probability theory. In these probabilistic models, the probability of a document d is relevant to a user query q is modelled explicitly [57] [60] [32]. Using relevance feedback to improve parameter estimation in probabilistic models is straightforward according to the definition of the models, because they presuppose relevance information.

In recent decades many researchers proposed IR models that are more general, while also explaining already existing IR models. For example, *Inference networks* have been successfully implemented in the well known INQUERY retrieval system [74]. Bayesian networks extend the view of inference networks. Both models represent documents and queries using acyclic graphs. Unfortunately, both models do not provide a sound theoretical framework to

learn the structure of the graph or to estimate the conditional probabilities defined on the graphs, and thus the model structure and parameter estimations are rather ad hoc [33]. Another example is the *language modeling approach*, which is a statistical approach that models the document generation process. This approach is a very active research area in the IR community since late 90's [27].

1.3.2 Existing adaptive filtering approaches

The key component of an adaptive filtering system is the user profile used by the system to make the decision of whether to deliver a document to the user or not. In the early research work as well as some recent commercial filtering systems, a user profile is represented as Boolean logic [36]. With the growing computation power and the advance of research in the information retrieval community in the last 20 years, filtering systems have gone beyond simple Boolean queries and represent a user profile as either a vector, a statistical distribution of words or something else. Much of the research on adaptive filtering is focused on learning a user profile from explicit user feedback on whether the user likes a document or not while interacting with the user. In general, there are two major approaches.

1.3.2.1 Filtering as Retrieval + thresholding

A typical retrieval system has a static information source, and the task is to return a ranking of documents in response to a short-term user request. Because of the influence of the retrieval models, some existing filtering systems use "retrieval scoring+thresholding" approach for filtering and build adaptive filtering based on algorithms originally designed for the retrieval task. A filtering system uses a retrieval algorithm to score each incoming document and delivers the document to the user if and only if the score is above a dissemination threshold. Some examples of retrieval models that have been applied to the adaptive filtering task are: Rocchio, language models, Okapi, and pseudo relevance feedback [3] [19] [48] [5] [26] [71].

A threshold is not needed in a retrieval task, because the system only needs to return a ranked list of documents. A major research topic in the adaptive filtering community is on how to set dissemination thresholds [64] [7] [82] [6] [91] [8] [17] [87]. The criteria of thresholds are often expressed in an easy to understand way, such as the utility function described in Section 1.2. At each time point, the system learns a threshold from the relevance judgements collected so far. For example, one direct utility optimization technique is to compute the utility on the training data for each candidate threshold, and choose the threshold that gives the maximum utility. Score distribution based approach assumes generative models of scores for relevant documents and non relevant documents. For example, one can assume the scores of relevant documents follow a Gaussian distribution, and the scores for none

relevant documents follow an exponential distribution. Training data can be used to estimate the model parameters, and the threshold can be found by optimizing the expected utility under the estimated model [7]. However, an adaptive filtering system only receives feedback for documents delivered/rated by the user, thus model estimation techniques based on random sampling assumption usually lead to biased estimation and should be avoided [91].

1.3.2.2 Filtering as text classification

Text classification is another well studied area. A typical classification system learns a classifier from a labeled training data set, and then classifies unlabeled testing documents into different classes. A popular approach is to treat filtering as a text classification task by defining two classes: relevant vs. non-relevant. The filtering system learns a user profile as a classifier and delivers a document to the user if the classifier thinks it is relevant or the probability of relevance is high. The state of the art text classification algorithms, such as support vector machines (SVM), K nearest neighbors (K-NN), neural networks, logistic regression and Winnow, have been used to solve this binary classification task [45] [20] [8] [62] [83] [90][71] [51] [80] [43] [72].

Instead of minimizing classification error, an adaptive filtering system needs to optimize the standard evaluation measure, such as a user utility. For example, in order to optimize the utility measure $T11U = 2R^+ - N^+$ (Equation 1.4), a filtering system usually delivers a document to the user if the probability of relevance is above 67% [61]. Some machine learning approaches, such as logistic regression or neural networks, estimate the probability of relevance directly, which makes it easier to make the binary decision of whether to deliver a document.

Many standard text classification algorithms do not work well for a new user, which usually means no or few training data points. Some new approaches have been developed for initialization. For example, researchers have found that retrieval techniques, such as Rocchio, work well at the early stage of filtering when the system has very few training data. Statistical text classification techniques, such as logistic regression, work well at the later stage of filtering when the system has accumulated enough training data. Techniques have been developed to combine different algorithms, and their results are promising [90]. Yet another example discussed in the following section is to initialize the profile of a new user based on training data from existing users.

It is worth mentioning that when adapting a text classification technique to the adaptive filtering task, one needs to pay attention that the classes are extremely unbalanced, because most documents are not relevant. The fact that the training data are not sampled randomly is also a problem that has not been well studied.

1.4 Collaborative adaptive filtering

One major challenge of building a recommendation or personalization system is that the profile learned for a particular user is usually of low quality when the amount of data from that particular user is small. This is known as the "cold start" problem. This means that any new user must endure poor initial performance until sufficient feedback from that user is provided to learn a reliable user profile.

There has been much research on improving classification accuracy when the amount of labeled training data is small. The semi-supervised learning approach combines unlabeled and labeled data together to achieve this goal [95]. Another approach is using domain knowledge. Researchers have modified different learning algorithms, such as Naïve-Bayes [46], logistic regression [28], and SVMs [81], to integrate domain knowledge into a text classifier. The third approach is borrowing training data from other resources [23] [28]. The effectiveness of these different approaches is mixed, due to how well the underlying model assumption fits the data.

This section introduces one well-received approach to improve recommendation system performance for a particular user is borrowing information from other users through a Bayesian hierarchical modeling approach. Several researchers have demonstrated that this approach effectively trades off between shared and user-specific information, thus alleviating poor initial performance for each user [96] [86] [93].

Assume there are M users in the adaptive filtering system. The task of the system is to deliver documents that are relevant to each user. For each user, the system learns a user model from the user's history. In the rest of this section, the following notations are used to represent the variables in the system.

$m = 1, 2, ..., M$**:** The index for each individual user. M is the total number of users.

w_m**:** The user model parameter associated with user m. w_m is a K dimensional vector.

$j = 1, 2, ..., J_m$**:** The index for a set of data for user m. J_m is the number of training data for user m.

$D_m = \{(x_{m,j}, y_{m,j})\}$**:** A set of data associated with user m. $x_{m,j}$ is a K dimensional vector that represents the mth user's jth training document.[6] $y_{m,j}$ is a scalar that represents the label of document $x_{m,j}$.

[6]The first dimension of x is a dummy variable that always equals to 1.

FIGURE 1.2: Illustration of dependencies of variables in the hierarchical model. The rating, y, for a document, x, is conditioned on the document and the user model, w_m, associated with the user m. Users share information about their models through the prior, $\Phi = (\mu, \Sigma)$.

$k = 1, 2, ..., K$**:** The dimensional index of input variable x.

The Bayesian hierarchical modeling approach has been widely used in real-world information retrieval applications. Generalized Bayesian hierarchical linear models, a simple Bayesian hierarchical models, are commonly used and have achieved good performance on collaborative filtering [86] and content-based adaptive filtering [96] [93] tasks. Figure 1.2 shows the graphical representation of a Bayesian hierarchical model. In this graph, each user model is represented by a random vector w_m. Assume a user model is sampled randomly from a prior distribution $P(w|\Phi)$. The system can predict the user label y of a document x given an estimation of w_m (or w_m's distribution) using a function $y = f(x, w)$. The model is called generalized Bayesian hierarchical linear model when $y = f(w^T x)$ is any generalized linear model such as logistic regression, SVM, and linear regression. To reliably estimate the user model w_m, the system can borrow information from other users through the prior $\Phi = (\mu, \Sigma)$.

Now we look at one commonly used model where $y = w^T x + \epsilon$, where $\epsilon \sim N(0, \sigma_\epsilon^2)$ is a random noise [86] [96]. Assume that each user model w_m is an independent draw from a population distribution $P(w|\Phi)$, which is governed by some unknown hyperparameter Φ. Let the prior distribution of user model w be a Gaussian distribution with parameter $\Phi = (\mu, \Sigma)$, which is the commonly used prior for linear models. $\mu = (\mu_1, \mu_2, ..., \mu_K)$ is a K dimensional vector that represents the mean of the Gaussian distribution, and Σ is the covariance matrix of the Gaussian. Usually, a Normal distribution $N(0, aI)$ and an Inverse Wishart distribution $P(\Sigma) \propto |\Sigma|^{-\frac{1}{2}b} \exp(-\frac{1}{2}c\text{tr}(\Sigma^{-1}))$ are used as hyperprior to model the prior distribution of μ and Σ respectively. I is the K dimensional identity matrix, and a, b, and c are real numbers.

With these settings, we have the following model for the system:

1. μ and Σ are sampled from $N(0, aI)$ and $IW_\nu(aI)$, respectively.

2. For each user m, w_m is sampled randomly from a Normal distribution: $w_m \sim N(\mu, \Sigma^2)$

3. For each item $x_{m,j}$, $y_{m,j}$ is sampled randomly from a Normal distribution: $y_{m,j} \sim N(w_m^T x_{m,j}, \sigma_\epsilon^2)$.

Let $\theta = (\Phi, w_1, w_2, ..., w_M)$ represent the parameters of this system that needs to be estimated. The joint likelihood for all the variables in the proba-

bilistic model, which includes the data and the parameters, is:

$$P(D,\theta) = P(\Phi)\prod_m P(w_m|\Phi)\prod_j P(y_{m,j}|x_{m,j},w_m) \tag{1.6}$$

For simplicity, we assume a, b, c, and σ_ϵ are provided to the system.

Researchers have shown that the Bayesian hierarchical modeling approach has a statistical significant improvement over the regularized linear regression model on several real world data sets. They observed a negative correlation between the number of training data for a user and the improvement the system gets. This suggests that the borrowing information from other users has more significant improvements for users with less training data, which is as expected. However, the strength of the correlation differs over data sets, and the amount of training data is not the only characteristics that will influence the final performance.

1.4.1 Computational consideration

One major concern about the hierarchical Bayesian modeling approach is the computation complexity. This problem has been addressed by exploiting the sparsity of the data space. A fast learning algorithm has been developed and tested on a real world data set (480,189 users, 159,836 features, and 100 million ratings). All the user models can be learned in about 4 hours using a single CPU PC(2GB memory, P4 3GHz) [93]. This demonstrates that the hierarchical Bayesian modeling technique can efficiently handle a large number of users and used in a large-scale commercial system.

1.5 Novelty and redundancy detection

Although there is an extensive body of research on adaptive information filtering, most of it is focused on identifying relevant documents. A common complaint about information filtering systems is that they do not distinguish between documents that contain new relevant information and documents that contain information that is relevant but already known. This is a serious problem, since a practical filtering system usually handles multiple document sources with significant amounts of redundant information. For example, a financial analyst only wants news stories that may affect the stock market, a market research analyst only wants new complaints about the product, and a news paper subscriber does not have time to read hundreds of similar news stories from different agencies about the same topic. In all these scenarios, topical relevancy is not enough because the users want *new* information. An information filtering system would provide better service to its users if it can filter out relevant documents that do not contain any new information.

The decision about whether a document contains new information depends on whether the relevant information in a document is covered by information in documents delivered previously. This complicates the filtering problem. The relevance of a document is traditionally a stateless Boolean value. A document either is or is not relevant, without regard as to where the document appears in the stream of documents. Decisions about redundancy and novelty depend very much on where in the stream a document appears.

Relevance and redundancy are significantly different concepts that require different solutions. A system that delivers documents that are novel and relevant must identify documents that are similar to previously delivered relevant documents in the sense of having a same topic, but also dissimilar to the previously delivered documents in the sense of containing new information. If the task is to deliver relevant documents, the learning algorithm will try to recognize documents similar to the delivered relevant documents (training data). Indeed, traditional evaluation of filtering systems (e.g., the TREC Adaptive Filtering track [63] [61] [62]) actually rewards systems for delivering redundant documents. If the task is to deliver only documents containing novel information, the learning algorithm must avoid documents that are similar to those already delivered. These two goals are in some sense contradictory, and it may be unrealistic to expect a single component to satisfy them both.

This suggests the redundancy problem needs a solution that's very different from the traditional adaptive information filtering models. We discuss some possible solutions in this section. We use the following notation throughout this section. All notation is defined with respect to a particular user profile.

- A, B: sets of documents
- d_t: a document that arrives at time t and that is being evaluated for redundancy.
- D_t: the set of all documents delivered for the profile by the time d_t arrives, not including d_t.
- d_j: usually refers to a relevant document that was delivered before d_t arrived.

When acquiring redundancy judgements and developing algorithms, we assume the redundancy of a new document d_t depends on the documents the user saw before d_t arrived. We also assume the documents the user saw before d_t arrived are the set of all documents delivered to the user profile by the time d_t arrives. We use $R(d_t) = R(d_t|D_t)$ to measure the redundancy of d_t.

One approach to novelty/redundancy detection is to cluster all previously delivered documents D_t, and then to measure the redundancy of the current document d_t by its distance to each cluster. This approach would be similar to solutions for the TDT First Story Detection problem [2]. This approach is sensitive to clustering accuracy, and is based on strong assumptions about the nature of redundancy.

Another approach is to measure redundancy based on the distance between the new document and each previously delivered document (document-document distance). This approach was developed by some researchers who argue that it may be more robust than clustering, and may be a better match to how users view redundancy. Because they found that it is easiest for user to identify a new document as being redundant with a single previously seen document, and harder to identify it as being redundant with *a set* of previously seen documents. The calculation of $R(d_t|D_t)$ is simplified by setting it equal to the value of the maximally similar value in all $R(d_t|d_j)$.

$$R(d_t|D_t) = max_{d_j \in D_t} R(d_t|d_j)$$

In the extreme case when d_t and d_j are exact duplicates ($d_t = d_j$), it is obvious that $R(d_t|d_j)$ should have a high value since a duplicate document is maximally redundant. One natural way to measure $R(d_t|d_j)$ is using measures of similarity/distance/difference between d_t and d_j.

One practical concern of redundancy estimation is the size of D_t could be very large. To reduce the computation cost during redundancy decisions, D_t can be truncated to the most recent documents delivered for a profile.

One possibly subtle problem characteristic is that redundancy is not a symmetric metric. d_j may cause d_k to be viewed as redundant, but if the presentation order is reversed, d_k and d_j may both be viewed as containing novel information. A simple example is a document d_k that is a subset (e.g., a paragraph) of a longer document d_j. This problem characteristic motivates exploration of asymmetric forms of traditional similarity/distance/difference measures.

Several different approaches to redundancy detection have been proposed and evaluated [92][4]. The simple set distance measure is designed for a Boolean, set based document models. The geometric distance (cosine similarity) measure is a simple metric designed for vector space document models. Several variations of KL divergence and related smoothing algorithms are more complex metrics designed to measure differences in probabilistic document models.

1.5.1 Set Difference

If each document is represented as a set of words, the set difference measure can be used to measure the redundancy of a new document. The novelty of a new document d_t is measured by the number of new words in the smoothed set representation of d_t. If a word w_k occurred frequently in document d_t but less frequently in an old document d_j, it is likely that new information not covered by d_j is covered by d_t.

Thus we can have the following measure for the novelty of the current document d_t with respect to an old document d_j.

$$R(d_t|d_j) = \|d_t \bigcap \overline{d_j}\| \tag{1.7}$$

We are not using the true difference between two sets

$$\|d_t \bigcap \overline{d_j}\| + \|\overline{d_t} \bigcap d_j\|$$

here because the words in

$$\|\overline{d_t} \bigcap d_j\|$$

shouldn't contribute to the novelty of d_t.

Different variations of the set represent of a document has been proposed. The simplest approach is to include a word in a set d_j if and only if the document contains the word. An alternative approach is to include a word in a set representation if an only if the number of times the word occurs in a document is larger than a threshold. However, some words are expected to be frequent in a new document because they tend to be frequent in the corpus, or because they tend to be frequent in all relevant documents. Stop words such as "the", "a", and "and" are examples of words that tend to be frequent in a corpus. There may also be topic-related stopwords, which are words that behave like stopwords in relevant documents, even if they are not stopwords in the corpus as a whole. To compensate for stop words, a third approach is to smooth a new document's word frequencies with word counts from *all* previously seen documents and word counts from *all delivered* (presumed relevant) documents [92].

1.5.2 Geometric Distance

If each document is represented as a vector, several different geometric distance measure, such as Manhattan distance and Cosine distance [44], can be used to measure the redundancy of a new document.

For example, prior research show that cosine distance, a symmetric measure related to the angle between two vectors [39], works reasonably well for redundancy detection. Represent d as a vector $d = (w_1(d), w_2(d), .., w_K(d))^T$, then:

$$R(d_t|d_j) = cosine(d_t, d_j) \tag{1.8}$$

$$= \frac{d_t \bullet d_j}{\mid d_t \mid\mid d_j \mid} \tag{1.9}$$

$$= \frac{\sum_{k=1}^{K} w_{k,t} w_{k,j}}{\sqrt{\sum_{k=1}^{K} w_{k,t}^2} \sqrt{\sum_{k=1}^{K} w_{k,j}^2}} \tag{1.10}$$

If we use $tf * idf$ score as the weight of each dimension of the document vector, we have $w_{k,j} = tf_{w_k,d_j} * idf_{w_k}$,

Where:

- $idf_{w_k} = \frac{log(\frac{C+0.5}{df_{w_k}})}{log(C+1.0)}$

- tf_{w_k,d_j} : the number of times word w_k occurs in document d_j
- df_{w_k} : the number of times word w_k occurs in documents the system processed
- C : the total number of document the system processed

1.5.3 Distributional Similarity

If each document is represented as a probabilistic document model, distribution similarity can be used to measure the redundancy of a new document. Probabilistic language models, which are widely used in speech recognition, have been very popular in information retrieval community over the last 10 years (e.g., [27]). The strong theoretical foundation of language models enables a variety of new capabilities, including redundancy detection. In the language model approach, a document is represented by a word distribution. Kullback-Leibler divergence, a distributional similarity measure, is a natural way to measure the redundancy of one document given another.

Representing document d as a unigram language model θ_d

$$R(d_t|d_j) = -KL(\theta_{d_t}, \theta_{d_j}) \tag{1.11}$$

$$= -\sum_{w_k} P(w_k|\theta_{d_t}) log(\frac{P(w_k|\theta_{d_j})}{P(w_k|\theta_{d_t})} \tag{1.12}$$

where θ_d is the language model for document d, and is a multinomial distribution.

θ_d can be found based on maximum likelihood estimation (MLE):

$$P(w_k|d) = \frac{tf_{w_k,d}}{\sum_{w_k} tf_{w_k,d}}$$

The problem with using MLE is that if a word never occurs in document d, it will get a zero probability ($P(w_k|d) = 0$). Thus a word in d_t but not in d_j will make $KL(\theta_{d_t}, \theta_{d_j}) = \infty$.

Smoothing techniques are necessary to adjust the maximum likelihood estimation so that the KL-based measure is more appropriate. Research shows that retrieval and filtering performance is highly sensitive to smoothing parameters when using language models. Several smoothing methods have been applied to ad-hoc information retrieval, text classification problems, and novelty detection [88][92].

1.5.4 Summary of novelty detection

The work described above is focused on the redundancy measure, and it is somewhat user independent in the sense that our redundancy measures only

calculate a score indicating the degree of redundancy in a document given a history of delivered documents. They do not actually make a decision as to whether a document is considered redundant or novel.

A **redundancy threshold** is needed in order to classify a document as redundant or novel. When human assessors are asked to make redundancy decisions given the same topics and document streams, they sometimes disagreed. In some cases the disagreement was based on differences in the assessors' internal definition of redundancy. However, more often one assessor might feel that a document d_t should be considered redundant if a previously seen document d_j covered 80% of d_t; the other assessor might not consider it redundant unless the coverage was more than 95%. A person's tolerance for redundancy can be modeled with a user-dependent threshold that converts a redundancy score into a redundancy decision. User feedback about which documents are redundant can serve as training data. Over time the system can learn to estimate the probability that a new document with a given redundancy score would be considered redundant. This probability can be expressed as $P(\text{user j thinks } d_t \text{ is redundant}|R(d_t|D_t))$.

1.6 Other adaptive filtering topics

While learning user profiles is an advantage of a filtering system, it is also a major research challenge in the adaptive filtering research community. Common learning algorithms require a significant amount of training data. However, a real-world filtering system must work as soon as the user uses the system, when the amount of training is extremely small or zero.[7] How should a good filtering system learn user profiles efficiently and effectively with limited user supervision while filtering? In order to solve this problem, researchers working on adaptive filtering have tried to develop robust learning algorithm that can work reasonably well when the amount of training data is small and more effective with more training data [85] [90]. Some filtering systems explore what the user likes while satisfying a user immediate information need and trade off exploration and exploitation [94] [22]. Some filtering systems consider many aspects of a document besides relevance, such as novelty, readability and authority [89] [84]. Some filtering systems use multiple forms of evidence, such as user context and implicit feedback from the user, while interacting with a user [89] [55].

This chapter does not cover all adaptive filtering topics in details due to the space limit and also because they are less "text" oriented. To finish this

[7] It is possible the system needs to begin working given a short user query and no positive instance.

section, some missed important topics are listed as follows, and the readers are referred to the papers cited for more details

1.6.1 Beyond bag of words

Most of the existing adaptive filtering approaches are focused on identifying relevant documents using distance measures defined in a document space indexed by text features such as keywords. This is a very simple and limited view of user modeling, without considering user context or other property of a document, such as whether a document is authoritative or whether it is novel to the user. However, even this simplest filtering task is still very hard, and existing filtering systems do not work effectively. Bayesian graphical modeling, a complex data driven user modeling approach, has been used to learn from implicit and explicit user feedback and to satisfy complex user criteria [89].

1.6.2 Using implicit feedback

For most of adaptive filtering work described in this section, we assume the system learns from explicit user feedback on whether a document delivered is relevant or not. There is much related work on using implicit feedback in the information retrieval community and the user modeling community. The work in these areas can be categorized according to the behavior category and minimum scope and have been reviewed recently [40]. There are many possible behaviors (view, listen, scroll, find, query, print, copy, paste, quote, mark up, type and edit) on different scope (segment, object and class) for system designers to choose from. Implicit feedback has also been explored for the task of filtering [54] [16] [53] [56] [89]. [54] suggested a list of potential implicit feedbacks. [16] build a personal news agent that used time-coded feedback from the user to learn a user profile. [53] investigated implicit feedback for filtering newsgroup articles.

1.6.3 Exploration and exploitation trade off

Most of the filtering systems deliver a document if and only if the expected immediate utility of delivering it is greater than the expected utility of not delivering it. However, delivering a document to the user has two effects: 1) it satisfies the user's information need immediately, and 2) it helps the system better satisfy the user in the future by learning from the relevance feedback about this document provided by the user. An adaptive information filtering approach is not optimal if it fails to recognize and model this second effect. Some researchers have followed this direction. [22] considers exploration benefit while filtering and carried out exploration and exploitation trade-off. [94] studies the second aspect and model the long term benefit of delivering a document as expected utility improvement as a result of improved model.

However, exploration and exploitation trade off is a problem far from being solved.

1.6.4 Evaluation beyond topical relevance

Utility is an approximation of the user's criteria of a good document. Given a utility measure, a system can achieve the objective of maximizing the user's satisfaction through utility maximization using mathematical or statistical techniques. A good utility measure is critical, because no system can do well with an inappropriate objective. In the IR community, utility is usually defined over relevance. Relevance was meant to represent a document's ability to satisfy the needs of a user. However, this concept is very abstract and hard to model, thus usually reduced to a narrow definition of "topical relevance" or "related to the matter at hand (aboutness)" [61] [76]. On the other hand, "presenting the documents in order of estimated relevance" without considering the incremental value of a piece of information is not appropriate [75]. Researchers have studied criteria such as information-novelty for retrieval [24], summarization [21], filtering [92] and topic detection and tracking [4]. Prior research on what is a user's perception/criteria have found that a wide range of factors (such as personal knowledge, topicality, quality, novelty, recency, authority and author qualitatively) affect human judgments of relevance [10] [49] [73] [78] [70]. We also discussed how to estimate novelty in this chapter, which is just an example of many of the important criteria for the user besides relevance, such readability [25] and authority [41]. How to build and evaluate a filtering system to optimize a more complex user criteria that goes beyond "topical relevance" or "aboutness" is still a challenging research problem for the adaptive filtering community.

1.7 ACKNOWLEDGEMENTS

The author would like to thank Jamie Callan, Thomas Minka, Yiming Yang, Wei Xu, Stephen Robertson, Chengxiang Zhai, James Allan, Sarah Tyler, Philip Zigoris, and Jonathan Koren for their contributions to the work reported in this chapter.

References

[1] James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. Topic detection and tracking pilot study. In *Topic Detection and Tracking Workshop Report*. 2001.

[2] James Allan, Victor Lavrenko, and Hubert Jin. First story detetion in tdt is hard. In *Proceedings of the 9th International Conference on Information and Knowledge Management*, pages 374–381, 2000.

[3] James Allan, Ron Papka, and Victor Lavrenko. On-line new event detection and tracking. In *roceedings of 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998.

[4] James Allan, Courtney Wade, and Alvaro Bolivar. Retrieval and novelty detection at the sentence level. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2003.

[5] A. Anghelescu, E. Boros, D. Lewis, V. Menkov, D. Neu, and P. Kantor. Rutgers filtering work at trec 2002: Adaptive and batch. In *Proceeding of the Eleventh Text REtrieval Conference (TREC-11)*, 2002.

[6] Avi Arampatzis. *Adaptive and Temporally-Dependant Document Filtering*. PhD thesis, Katholieke Universiteit Nijmegen, Nijmegen, Netherland, 2001.

[7] Avi Arampatzis and A. Hameren. The score-distribution threshold optimization for adaptive binary classification task. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 285–293, 2001.

[8] Thomat Ault and Yiming Yang. knn, rocchio and metrics for information filtering at trec-10. In *Proceeding of the Tenth Text REtrieval Conference (TREC-10)*. National Institute of Standards and Technology, special publication 500-225, 2001.

[9] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[10] C.L. Barry. User-defined relevance criteria: An exploratory study. In *Journal of the American Society for Information Science, 45(3)*, 1994.

0-8493-0052-5/00/$0.00+$.50

[11] C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 1998.

[12] C. Basu, H. Hirsh, W. Cohen, and C. Nevill-Manning. Technical paper recommendation: A study in combining multiple information sources. *Journal of Artificial Intelligence Research*, 2001.

[13] Steven M. Beitzel, Eric C. Jensen, Abdur Chowdhury, David Grossman, and Ophir Frieder. Evaluation of filtering current news search results. In *SIGIR '04: Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 494–495. ACM Press, 2004.

[14] N. Belkin and B. Croft. Information filtering and information retrieval: two sides of the same coin? In *Communications of the ACM*, 1992.

[15] Robert Bell, Yehuda Koren, and Chris Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 95–104, New York, NY, USA, 2007. ACM Press.

[16] Daniel Billsus and Michael J. Pazzani. A personal news agent that talks, learns and explains. In *AGENTS '99: Proceedings of the third annual conference on Autonomous Agents*, pages 268–275. ACM Press, 1999.

[17] M. Boughanem, H. Tebri, and M. Tmar. IRIT at TREC 2002: filtering track. In *Proceeding of the Eleventh Text REtrieval Conference (TREC-11)*, 2002.

[18] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. Technical report, Microsoft Research, One Microsoft Way, Redmond, WA 98052, 1998.

[19] Jamie Callan. Document filtering with inference networks. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 262–269, 1996.

[20] Nicola Cancedda, Nicolo Cesa-Bianchi, Alex Conconi, Claudio Gentile, Cyril Goutte, Thore Graepel, Yaoyong Li, Jean Michel Renders, John Shawe Taylor, and Alexei Vinokourov. Kernel method for document filtering. In *The Eleventh Text REtrieval Conference (TREC11)*. National Institute of Standards and Technology, special publication 500-249, 2003.

[21] Jaime Carbonell and Jade Goldstein. Automatic text summarization of multiple documents. In *Proceedings of the 21th Annual International*

ACM SIGIR Conference on Research and Development in Information Retrieval, 1998.

[22] Kian Ming Adam Chai, Hai Leong Chieu, and Hwee Tou Ng. Bayesian online classifiers for text classification and filtering. In *Proceedings of 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2002.

[23] Ciprian Chelba and Alex Acero. Adaptation of maximum entropy capitalizer: Little data can help a lot. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 285–292, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[24] Harr Chen and David R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 429–436, New York, NY, USA, 2006. ACM Press.

[25] K. Collins-Thompson and J. Callan. Predicting reading difficulty with statistical language models. In *Submitted to Journal of the American Society for Information Science and Technology.*, 2003.

[26] K. Collins-Thompson, P. Ogilvie, Y Zhang, and J. Callan. Information filtering, novelty detection, and named-page finding. In *Proceeding of the Eleventh Text REtrieval Conference (TREC-11)*, 2002.

[27] Bruce Croft and John Lafferty, editors. *Language Modeling for Information Retrieval*. Kluwer, 2002.

[28] Aynur Dayanik, David D. Lewis, David Madigan, Vladimir Menkov, and Alexander Genkin. Constructing informative prior distributions from domain knowledge in text classification. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 493–500, New York, NY, USA, 2006. ACM Press.

[29] J. Delgado and N. Ishii. Memory-based weighted majority prediction for recommender systems. In *ACM SIGIR'99 Workshop on Recommender Systems*, 1999.

[30] Ugur etintemel, Michael J. Franklin, and C. Lee Giles. Self-adaptive user profiles for large-scale data delivery. In *ICDE '00: Proceedings of the 16th International Conference on Data Engineering*, page 622, Washington, DC, USA, 2000. IEEE Computer Society.

[31] Christos Faloutsos and Douglas W. Oard. A survey of information retrieval and filtering methods. Technical report, Univ. of Maryland, College Park, 1995.

[32] N. Fuhr. Probabilistic models in information retrieval. In *The Computer Journal*, volume 35(3), pages 243–255, 1992.

[33] Robert Fung and Brendan Del Favero. Applying bayesian networks to information retrieval. *Communications of the ACM*, 38(3):42–ff., 1995.

[34] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, New York, NY, USA, 1999. ACM Press.

[35] Thomas Hofmann and Jan Puzicha. Latent class models for collaborative filtering. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 688–693, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[36] Edward M Housman. Selective dissemination of information. In Carlos A Cuandra, editor, *Annual Review of Information Science and Technology. Vol. 8. American Socienty for Information Science*, 1973.

[37] Rong Jin, Joyce Y. Chai, and Luo Si. An automatic weighting scheme for collaborative filtering. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 337–344, New York, NY, USA, 2004. ACM Press.

[38] Karen Sparck Jones, Steve Walker, and Stephen E. Robertson. A probabilistic model of information retrieval: development and comparative experiments - part 2. *Information Processing and Management*, 36(6):809–840, 2000.

[39] William P. Jones and George W. Furnas. Pictures of relevance. *Journal of the American Society for Information Science*, 1987.

[40] Diane Kelly and Jaime Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28, 2003.

[41] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[42] Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

[43] K.-S. Lee, K. Kageura, and A. Aizawa. TREC 11 experiments at NII: The effects of virtual relevant documents in batch filtering. In *Proceeding of the Eleventh Text REtrieval Conference (TREC-11)*, 2002.

[44] Lillian Lee. Measures of distributional similarity. In *Proceedings of the 37th ACL*, 1999.

[45] David Lewis. Applying support vector machines to the TREC-2001 batch filtering and routing tasks. In *Proceeding of the Eleventh Text REtrieval Conference (TREC-11)*, 2002.

[46] Bing Liu, Xiaoli Li, Wee Sun Lee, , and Philip Yu. Text classification by labeling words. In *Proceedings of The Nineteenth National Conference on Artificial Intelligence (AAAI-2004)*, July 25-29, 2004.

[47] Robert M. Losee and Abraham Bookstein. Integrating boolean queries in conjunctive normal form with probabilistic retrieval models. In *Information Processing and Management*, 1988.

[48] L. Ma, Q. Chen, S. Ma, M. Zhang, and L. Cai. Incremental learning for profile training in adaptive document filtering. In *Proceeding of the Eleventh Text REtrieval Conference (TREC-11)*, 2002.

[49] K.L. Maglaughlin and D.H. Sonnenwald. User perspectives on relevance criteria: A comparison among relevant, partially relevant, and not-relevant judgments. In *Journal of the American Society for Information Science and Technology*, 2003.

[50] A. Martin, G. Doddington, T. Kamm, and M. Ordowski. The DET curve in assessment of detection task performance. In *Proceedings of EuroSpeech*, 1997.

[51] P. McNamee, C. Piatko, and J. Mayfield. JHU/APL at TREC 2002: Experiments in filtering and arabic retrieval. In *Proceeding of the Eleventh Text REtrieval Conference (TREC-11)*, 2002.

[52] Prem Melville, Raymond J. Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002)*, Edmonton, Canada, 2002.

[53] Masahiro Morita and Yoichi Shinoda. Information filtering based on user behavior analysis and best match text retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 272–281. Springer-Verlag New York, Inc., 1994.

[54] D. M. Nichols. Implicit rating and filtering. In *Proceedings of the Fifth DELOS Workshop on Filtering and Collaborative Filtering*, 1997.

[55] Doug Oard and Jinmook Kim Contact. User modeling for information access based on implicit feedback, 2001.

[56] D.W. Oard and J. Kim. Modeling information content using observable behavior. In *ASIST 2001 Annual Meeting*.

[57] Van Rijbergen and J.C. A theoretical basis for the use of co-occurrence data in information retrieval. In *Journal of Documentation*, pages 106–119, 1976.

[58] V. Rijsbergen. *Information Retrieval*. 1979.

[59] S. Robertson and D.A. Hull. The TREC-9 Filtering track report. In *The Ninth Text REtrieval Conference (TREC-9)*, pages 25–40. National Institute of Standards and Technology, special publication 500-249, 2001.

[60] S. Robertson and K. Sparck Jones. Relevance weighting of search terms. In *Journal of the American Society for Information Science*, volume 27, pages 129–146, 1976.

[61] S. Robertson and I. Soboroff. The TREC-10 Filtering track final report. In *Proceeding of the Tenth Text REtrieval Conference (TREC-10)*, pages 26–37. National Institute of Standards and Technology, special publication 500-250, 2002.

[62] S. Robertson and I. Soboroff. The TREC 2002 filtering track report. In *Proceeding of the Eleventh Text REtrieval Conference (TREC-11)*, 2002.

[63] S. Robertson and S. Walker. Microsoft Cambridge at TREC-9: Filtering track. In *Proceeding of Ninth Text REtrieval Conference (TREC-9)*, pages 361–368. National Institute of Standards and Technology, special publication 500-249, 2001.

[64] S. Robertson and Stephen Walker. Threshold setting in adaptive filtering. *Journal of Documentation*, pages 312–331, 2000.

[65] Stephen Robertson. On theoretical argument in information retrieval. Salton Award Lecture given at SIGIR 2000, July 2000.

[66] J. J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System– Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, 1971.

[67] R. R. Salakhutdinov, A. Mnih, and G. E. Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceeding of the International Conference on Machine Learing (ICML 2007)*, 2007.

[68] G. Salton and C. Buckley. *Term-weighting approaches in automatic text retrieval*, volume 24. 1988.

[69] G. Salton and M. McGill. *Introduction to Modern Informatin Retrieval*. McGraw-Hill, 1983.

[70] Linda Schamber and Judy Bateman. User criteria in relevance evaluation: Toward development of a measurement scale. In *ASIS 1996 Annual Conference Proceedings*, October 1996.

[71] M. Srikanth, X. Wu, and R. Srihari. UB at TREC 11: Batch and adaptive filtering. In *Proceeding of the Eleventh Text REtrieval Conference (TREC-11)*, 2002.

[72] M. Stricker, F. Vichot, G. Dreyfus, and F. Wolinski. Training context-sensitive neural networks with few relevant examples for the TREC-9 routing. In *The Ninth Text REtrieval Conference (TREC9)*. National Institute of Standards and Technology, special publication 500-249, 2000.

[73] Anastasios Tombros, Ian Ruthven, and Joemon M. Jose. How users assess web pages for information seeking. *J. Am. Soc. Inf. Sci. Technol.*, 56(4):327–344, 2005.

[74] Howard R. Turtle. *Inference Networks for Document Retrieval*. PhD thesis, University of Massachusetts, October 1990.

[75] H. R. Varian. Economics and search (invited talk at SIGIR 1999), 1999.

[76] E. M. Voorhees and Lori P. Buckland, editors. *NIST Special Publication 500-251: The Eleventh Text REtrieval Conference (TREC 2002)*. Department of Commerce, National Institute of Standards and Technology, 2002.

[77] Jun Wang, Arjen P. de Vries, and Marcel J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 501–508, New York, NY, USA, 2006. ACM Press.

[78] Peiling Wang. *A cognitive model of document selection of real users of IR Systems*. PhD thesis, University of Maryland, 1994.

[79] Charles L. Wayne. Multilingual topic detection and tracking: Successful research enabled by corpora and evaluation. In *LREC*, 2000.

[80] L. Wu, X. Huang, J. Niu, Y. Xia, Z. Feng, and Y. Zhou. FDU at TREC 2002: Filtering, Q&A, web and video tasks. In *Proceeding of the Eleventh Text REtrieval Conference (TREC-11)*, 2002.

[81] X. Wu and R. K. Srihari. Incorporating prior knowledge with weighted margin support vector machines. In *Proc. ACM Knowledge Discovery Data Mining Conf.(ACM SIGKDD 2004)*, Aug. 2004.

[82] Y. Yang and B. Kisiel. Margin-based local regression of adaptive filtering. In *Proceedings of the Twelveth International Conference on Information Knowledge Management (CIKM 2003)*. ACM Press, 2003.

[83] Y. Yang, S. Yoo, J. Zhang, and B. Kisiel. Robustness of adaptive filtering methods in a cross-benchmark evaluation. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2005.

[84] Yiming Yang, Abhimanyu Lad, Ni Lao, Abhay Harpale, Bryan Kisiel, and Monica Rogati. Utility-based information distillation over temporally sequenced documents. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 31–38, New York, NY, USA, 2007. ACM Press.

[85] Kai Yu, Volker Tresp, and Anton Schwaighofer. Learning gaussian processes from multiple tasks. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 1012–1019, New York, NY, USA, 2005. ACM Press.

[86] Kai Yu, Volker Tresp, and Shipeng Yu. A nonparametric hierarchical bayesian framework for information filtering. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 353–360. ACM Press, 2004.

[87] C. Zhai, P. Jansen, and E. Stoica. Threshold calibration in CLARIT adaptive filtering. In *Proceeding of Seventh Text REtrieval Conference (TREC-7)*, pages 149–157. National Institute of Standards and Technology, special publication 500-242, 1999.

[88] Chengxiang Zhai and John Lafferty. A study of smoothing methods for langauge models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 334–342, September 2001.

[89] Y. Zhang and J. Callan. Combine multiple forms of evidence while filtering. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005.

[90] Yi Zhang. Using Bayesian priors to combine classifiers for adaptive filtering. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004.

[91] Yi Zhang and Jamie Callan. Maximum likelihood estimation for filtering thresholds. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 294–302, 2001.

[92] Yi Zhang, Jamie Callan, and Tom Minka. Novelty and redundancy detection in adaptive filtering. In *Proceedings of the 25th ACM SIGIR Conference*, 2002.

[93] Yi Zhang and Jonathan Koren. Efficient bayesian hierarchical user modeling for recommendation systems. In *Proceedings of the 30st Annual*

International ACM SIGIR Conference on Research and Development in Information Retrieval, 2007.

[94] Yi Zhang, Wei Xu, and Jamie Callan. Exploration and exploitation in adaptive filtering based on bayesian active learning. In *Proceeding of the International Conference on Machine Learing (ICML 2003)*, 2003.

[95] Xiaojin Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin – Madison, December 9, 2006.

[96] Philip Zigoris and Yi Zhang. Bayesian adaptive user profiling with explicit & implicit feedback. In *Conference on Information and Knowledge Mangement 2006*, 2006.