Bayesian Adaptive User Profiling with Explicit & Implicit Feedback

Philip Zigoris
Department of Computer Science
University of California, Santa Cruz
zigoris@soe.ucsc.edu

Yi Zhang
School of Engineering
University of California, Santa Cruz
yiz@soe.ucsc.edu

ABSTRACT

Research in information retrieval is now moving into a personalized scenario where a retrieval or filtering system maintains a separate user profile for each user. In this framework, information delivered to the user can be automatically personalized and catered to individual user's information needs. However, a practical concern for such a personalized system is the "cold start problem": any user new to the system must endure poor initial performance until sufficient feedback from that user is provided.

To solve this problem, we use both explicit and implicit feedback to build a user's profile and use Bayesian hierarchical methods to borrow information from existing users. We analyze the usefulness of implicit feedback and the adaptive performance of the model on two data sets gathered from user studies where users' interaction with a document, or *implicit feedback*, were recorded along with explicit feedback. Our results are two-fold: first, we demonstrate that the Bayesian modeling approach effectively trades off between shared and user-specific information, alleviating poor initial performance for each user. Second, we find that implicit feedback has very limited unstable predictive value by itself and only marginal value when combined with explicit feedback.

Keywords

Information Retrieval, User Modeling, Bayesian Statistics, Implicit Feedback

1. INTRODUCTION

Although ad hoc retrieval systems have become part of the daily life of internet users or digital library users, it is clear that there is great variety in users' needs. Thus, such systems can not offer the best possible service since they are not tailoring information to individual user needs.

IR research is now moving into a more complex environment with user centered or personalized adaptive information retrieval, information filtering and recommendation systems as major research topics. As opposed to traditional ad hoc systems, a personalized system adaptively learns a profile for each user, automatically catering to the user's specific needs and preferences [1].

To learn a reliable user specific profile, an adaptive system usually needs a significant amount of explicit feedback (training data) from the user. However, the average user doesn't like to answer a lot of questions or provide explicit feedback on items they have seen. Meanwhile, a user does not want to endure poor performance while the system is "training" and expects a system to work reasonably well as soon as he/she first uses the system. Good initial performance is an incentive for the user to continue using the system. Thus an important aspect of personalization is to develop a system that works well initially with less explicit user feedback.

Much prior research has been carried out exploring the usefulness of implicit feedback [10] because it is easy to collect and requires no extra effort from the user. On the other hand, user independent systems perform reasonably well for most users, mainly because the system parameters have been tuned to a reasonable value based on thousands of existing users. These observations suggest at least two possible approaches to improve the early stage performance of a personalized system: using cheap implicit feedback from the user and borrowing information from other users. Both approaches may help the system reduce its uncertainty about the user, especially when the user just starts using the system and has not provided much explicit feedback.

This paper explores both directions under a single, unified Bayesian hierarchical modeling framework. Intuitively, combining the two approaches under this framework may achieve a nice tradeoff between bias and variance trade off, especially for a new user. This is because including implicit feedback decreases the bias and increases variance of the learner, while borrowing information from other users through a Bayesian prior in our framework has the inverse effect. We demonstrate, empirically, that the hierarchical model controls the tradeoff between shared and personal information, thereby alleviating poor initial performance. We also evaluate the long-term usefulness of different types of feedback for predicting a user's rating for a document.

The paper is organized as follows. We begin with a brief review of related work in Section 2. We then describe the Bayesian hierarchical modeling approach to learn user specific models (Section 3) and introduce a computationally efficient model, the Hierarchical Gaussian network, as an example to be used in our experiments (Section 4). Section

CIKM'06, November 5–11, 2006, Arlington, Virginia, USA.

5 and 6 describe the experimental methodology and results on two real user study data set. Section 7 concludes.

2. RELATED WORK

Implicit feedback is a broad term including any kind of natural interaction of a user with a document [16]. Examples of implicit feedback are mouse and keyboard usage, page navigation, book-marking, and editing. While the focus of much emerging research, there is still some debate over the value of implicit feedback. For example, it has been observed in numerous controlled studies [3, 12, 6, 4] that there is a clear correlation between the time a user spends viewing a document and how useful they found that document. However, Kelly, et al. [9] demonstrate, in a more naturalistic setting, that this correlation varies significantly across tasks and conclude that the usefulness of implicit feedback is conditioned heavily on what type of information the user is seeking. Standard machine learning techniques such as SVMs [8], neural networks [12], and graphical models [4][18] have been used to explore the usefulness of implicit feedback, with varying degrees of success. In our work we use a hierarchical Bayesian hierarchical model to integrate implicit feedback.

The idea of borrowing information from other users to serve the information needs of a user is widely studied in the area called collaborative filtering and Bayesian hierarchical modeling has applied in this context [17]. The major differences in our work are: 1) we focus on developing complex user models that go beyond relevance based content model by including other explicit feedback (such as topic familiarity and readability) and implicit feedback (such as user actions and system context); 2) we focus on the *adaptive* learning environment and analyze the online performance of the learning system. This analysis makes it clear how an *adaptive* system can benefit from the hierarchical Bayesian framework; and 3) we use a Gaussian network (Gaussian user models with Gaussian prior), while the prior work uses different functional forms, such as multinomial user models with Dirichlet prior [17]. Our method is more computationally efficient and also has less privacy concerns while sharing information across users.

3. HIERARCHICAL BAYESIAN FRAMEWORK

One concern about using user-specific models, or profiles, is that there is an initial period where the user must endure poor performance. On the other hand, existing user independent systems seem to work well by tuning parameters for the general public. This motivates us to borrow information from existing users to serve a new user. Here we adopt a principled approach for doing this based on Bayesian hierarchical models.

Let f^u represent the model of a user u. Over time, the user interacts with the system and provides explicit and implicit feedback about documents the user has read. Let a d-dimensional random variable x^u represent the information about a document for user u. Each dimension of x^u corresponds to a feature, and the features could be user actions on the document, the document relevance scores the system derives from the user's explicit feedback on other documents, user-independent features of the document, and so on. The user u may provide a rating y^u to a document[1] x^u according to the user model f^u. For now, a *model* is a function that takes information about a document and returns an estimation of whether the user likes the document or not (rating): $f^u : x^u \rightarrow y^u$. The functional form for f^u varies for different learning system, and we delay this discussion until Section 4. We make no assumption about the distribution of documents.

A Bayesian based learning system begins with a certain prior belief, $P(f|\theta)$, about the distribution of user models. In the simplest terms the hierarchical Bayesian framework can be written as

$$\begin{aligned} f^u &\sim P(f|\theta) \\ y &= f^u(x) \end{aligned}$$

and is also illustrated in Figure 1 Note that this is a very general framework and can accommodate any class of functions for which a reasonable prior can be specified. This includes any function parameterized by real number (e.g. SVMs, neural networks with fixed topology) and regression trees [2].

A personalized system tries to learn the user model f^u for a user u. As the user uses the system, the system receives a sample of document-rating pairs, $D_u = \{(x_i^u, y_i^u)|i = 1 \dots N_u\}$, where N_u is the number of training examples for user u. Using Baye's Rule, the system can update its belief about the user model based on the data and get the *posterior* distribution over user models:

$$\begin{aligned} P(f^u|\theta, D_u) &= \frac{P(D_u|f^u, \theta)P(f^u|\theta)}{P(D_u|\theta)} \\ &= P(f^u|\theta)\prod_{i=1}^{N_u}\frac{P(f^u(x_i^u) = y_i^u|f^u)}{P(f^u(x_i^u) = y_i^u|\theta)} \end{aligned}$$

To find the maximum a posteriori (MAP) model, f_{MAP}^u, we can ignore the denominator since it is the same for all models. Then we have:

$$\begin{aligned} f_{MAP}^u &= \text{argmax}_f \quad P(f|\theta, D_u) \\ &= \text{argmax}_f \quad P(f|\theta)\prod_{i=1}^{N_u} P(f(x_i^u) = y_i^u|f) \\ &= \text{argmax}_f \quad \log P(f|\theta) + \\ &\qquad \sum_{i=1}^{N_u} \log P(f(x_i^u) = y_i^u|f) \end{aligned} \tag{1}$$

Equation 1 shows clearly how the fitness of a model is decomposed into the model's prior probability and the data likelihood given the model.

Incorporating a prior distribution automatically manages the trade off between generic user information and user specific information based on the amount of training data available for the specific user. When a user u starts using a system, the system has almost no information about the user. Because N_u for the user is small, the prior distribution $p(f|\theta)$ is a major contributor when estimating the user model $\hat{f}$ (first term in Equation 1). If the system has a reliable prior, it can perform well even with little data for the user. How can we find a good prior for f? We treat

[1]This paper sometimes refers to x^u as simply a "document" even though it has features of the user's interaction with it.

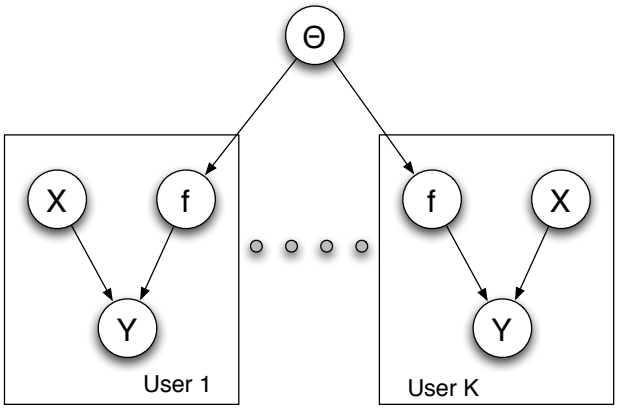

Figure 1: Illustration of dependencies of variables in the hierarchical model. The rating, Y, for a document, X, is conditioned on the document and the model, W, associated with that user. Users share information about their model through the use of a prior, Θ.

user u's model f^u as a sample from the distribution $P(f|\theta)$. Given a set of user models, we can choose θ to maximize their likelihood.

As the system gets more training data for this particular user, f^u_{MAP} is dominated by the data likelihood (second term in Equation 1) and the prior learned from other users becomes less important. As a result, the system works like a well tuned non-personalized system for a new user, but keeps on improving the user model as more feedback from the user is available. Eventually, after a sufficient amount of data is collected, the user's profile should reach a fixed point.

4. HIERARCHICAL GAUSSIAN NETWORK

Specific functional forms for f^u and $P(f|\theta)$ are needed to build a practical learning system. For our work, we let f be a simple linear regression function and let w be the parameter of f. We chose to represent the prior distribution as a multivariate normal, i.e. $P(f|\theta) = P(w|\theta) = N(w; \mu, \Sigma)$. The motivation for this choice is the fact that the self-conjugacy of the normal distribution simplifies computation. For simplicity, we assume that the covariance matrix Σ is diagonal, i.e. the components of the model are independent. Thus we have:

$$\begin{aligned} w^u &\sim \mathcal{N}(w; \mu, \Sigma) \\ y &= x^T \cdot w^u + \varepsilon \\ \varepsilon &\sim \mathcal{N}(\varepsilon; 0, \kappa_u^2) \end{aligned}$$

where ε is independent zero mean Gaussian noise. One can also view y as a random sample from a normal distribution $N(y; x^T \cdot w^u, \kappa_u^2)$. It is possible to extend the hierarchical model to include a prior on the variance of the noise κ^u and learn it from the data. For simplicity, we set κ_u^2 to a constant value of 0.1 for all u in our experiments.

Based on Equation 1, the MAP estimate of the user model w^u for a particular user u is [7]:

$$\begin{aligned} w^u_{MAP} &= \text{argmax}_w \quad \log P(w|\theta) + \sum_{i=1}^{N_u} \log P(x_i^u, y_i^u|w) \\ &= \text{argmin}_w \quad (w-\mu)^T \Sigma^{-1} (w-\mu) + \\ &\qquad \frac{1}{\kappa_u} \sum_{i=1}^{N_u} (x_i^T \cdot w - y_i)^2 \end{aligned} \tag{2}$$

Equation 2 also shows the natural tradeoff that occurs between the prior and the data. As we collect more data for user u, the second term is expected to grow linearly with N_u. This term, which corresponds to the mean square error (L_2 loss) of the model w on the training sample, will eventually dominate the first term, which is the loss associated with the deviation from the prior mean.

The variance of the prior Σ also affects this tradeoff. For instance, if the values on the diagonal of Σ are very large for all j, the prior is close to a uniform distribution and w_{MAP} is close to the linear least square estimator. In the reverse situation, the prior dominates the data and the MAP estimate will be closer to μ.

So far, we have discussed the learning process of a single user and assume the prior is fixed. However, the system will have many users, and the prior should be learned from the other users data. We use an initially uniform prior when the system is launched [2], i.e. $P(\mu, \Sigma)$ is the same for all μ, Σ.

As users join the system, the prior distribution $p(f|\theta)$ is updated. At certain point when the system has K existing users, the system has a set of K user models w^u: $u = 1 \ldots K$. Based on the evidence from these existing users, we can infer the parameters $\theta = (\mu, \Sigma)$ of our prior distribution:

$$\hat{\mu} = \frac{1}{k} \sum_{u=1}^{K} w^u \tag{3}$$

$$\hat{\Sigma} = \frac{1}{k-1} \sum_{u=1}^{K} (w^u - \hat{\mu}) \cdot (w^u - \hat{\mu})^T \tag{4}$$

This is, in essence, the mechanism by which we share user information: the prior mean is the average of that parameter for all other user models. Similarly, the prior variance captures how consistent a parameter is for all users.

4.1 Finding w_{MAP} with Belief Propagation

Given a large pool of data, we need to find the MAP estimation of all f^u and θ in Figure 1. A simple approach is to treat it as one large optimization problem where all parameters are optimized simultaneously. If there are K users with data sets $\bar{D} = \{D_1, \ldots, D_K\}$ then we can write down the objective function as:

$$\begin{aligned} P(\bar{w}, \mu, \Sigma|\bar{D}) &= \frac{P(\bar{D}|\bar{w}) P(\bar{w}|\mu, \Sigma) P(\mu, \Sigma)}{P(\bar{D})} \\ &\propto P(\bar{D}|\bar{w}) P(\bar{w}|\mu, \Sigma) P(\mu, \Sigma) \\ &\propto \prod_{u=1}^{K} P(D_u|w_u) P(w_u|\mu, \Sigma) \\ &\propto \prod_{u=1}^{K} P(w_u|\mu, \Sigma) \prod_{i=1}^{N_u} P(y_i^u|w_u, x_i^u) \end{aligned} \tag{5}$$

[2]It is, of course, possible to extend the hierarchical model to include a prior distribution for the parameters μ and Σ.

where $\bar{w} = \{w^1, \ldots, w^K\}$. This objective function corresponds to the posterior distribution over user models and the parameters of the prior distribution; it is the product of the posterior distributions of each user model, but with μ and Σ also maintained as free variables. Importantly, this function is concave and standard optimization algorithms exist for finding its global maxima. The problem with this approach, from a system's perspective, is that we cannot update one user's model without updating everyone else's. Fortunately, a user's model is independent of other users' models given the prior θ, and the model prior θ learned from many existing users is unlikely to change much when a new user joins the system. Message passing algorithms, or *belief propagation*, take advantage of this independence. Specifically, if we assume θ is fixed then we can update w^u independently of any other user's data.

Although privacy is not a major focus of this paper, this learning process alleviates a common concern with sharing data among users. In a distributed environment where each individual user/peer keeps his/her own data and user model, w^u is the sufficient statistic summarizing all the data about a user u, and only this information needs to be propagated to a central server that keeps the prior. The central server aggregates all the user models and summarizes them as (μ, Σ) and shares only this information with each individual user.

5. EXPERIMENTAL DESIGN

5.1 Experimental data sets

Some experiments are carried out to understand the Bayesian hierarchical user modeling approach using Gaussian networks and value of implicit feedback. The following two data sets are used in our experiments:

[3] Claypool, et al. developed a web browser (called Curious Browser) that records various types of implicit feedback. A group of 75 student volunteers were instructed to browse the web for 20-30 minutes and provided explicit ratings for each document visited. The browsing task is unrestricted and unguided, allowing for a variety of tasks. The types of implicit feedback include aspects of mouse usage (time moving, scroll bar clicks), keyboard usage (arrow key and page up-down clicks/time), and the time spent viewing the page. Explicit feedback is limited to the user's rating. In total, explicit ratings for 1823 documents were gathered.

[18] Zhang modified the Curious Browser and used it for a different user study where more forms of explicit feedback are collected. Participants in this study were focused specifically on reading news articles from different news feeds. The study lasted for one month and participants were required to spend at least one hour every day using the system. The implicit feedback on this data set is the same as Claypool's. A set of user-independent features ("document length", "the number of pages linking to the news feeds host server", "the speed of the host server") are also collected. Users in this study also provided explicit feedback on other aspects of the document such as readability and topic familiarity, thus the input variable x includes three features the system derived from the explicit feedback ("relevance score", "readability score", "topic familiar before"). The system derived features help the system integrate a traditional IR-type approach that relies heavily on document content into the user modeling framework. All samples with missing values are removed as well as users with fewer than 50 samples. In total, there were 15 users and explicit feedback for 4663 documents.

name	source	I	D
Z_D	[18]		✓
Z_{ID}	[18]	✓	✓
Z_I	[18]	✓	
C	[3]	✓	

Table 1: Listing of the datasets used in the experimental evaluation. The third (I) and fourth (D) columns indicate whether the dataset contains implicit and explicit feedback, respectively.

In both studies the users rated the documents on a scale of 1 to 5; a rating of 1 indicated the document was not of interest and a rating of 5 indicated it was.

All the features were normalized to zero mean and unit variance. These two data sets were collected through two user studies carried out by different research groups. They enable us to evaluate our tasks on data collected in very different experimental settings. There are no queries in both tasks. Compared to the traditional evaluation data set used in TREC, these data sets are very noisy and many of the features are not very informative.

In order to evaluate the usefulness of different types of features we formed 4 data sets, listed in Table 1 in total by taking subsets of features. Let D be the set of features the system gets right after the document arrives and before the user reads the document. This includes three features the system derived from the explicit feedback ('relevance score", "readability score", "topic familiar before") as well as the document length, server speed, and number of in-links to the news server. Since the first three features in D are learned from the user's explicit feedback, we also refer to D as explicit feedback. Let I refer to implicit feedback acquired after the user reads the document. Claypool's study included only features in set I and Zhang's study included both. It is important to note that our use of implicit feedback departs from a realistic application setting: we can't very well decide to deliver a document based on the user's interaction with it. This information is only available after the document has been delivered. Regardless, it is worthwhile to explore the predictive abilities of such a cheap resource.

5.2 Evaluation methodology

Our aim is to see how a system that has already established itself can accommodate a new user, not to emulate a system starting from scratch. For each user, we simulate the following setting: all other users are already present on the system and *then* the user in question begins to use the system. This approach is very similar to the "leave one out" technique, and each time we leave one user out to test the system perform on that particular user over time. The original order in which the user viewed the documents is preserved. The system receives the user rating right after the user finishes reading a document, and this training example

is used to immediately update the user's model.

The experiments compare the following four separate models (Following each model name is its corresponding symbol in the experimental results figures that we will discuss in Section 6):

Prior (◯) Model based on the hierarchical framework described in Section 4. All other user's models are assumed stable and are not updated while learning and evaluating the adaptive performance of one particular user in the simulated study. Using belief propagation and *all* the data available for other users, the system estimate the prior distribution.

No Prior (◇) Linear model trained with only this user's data. This model is effectively equal to having a prior with an infinitely large variance.

Generic (∗) Linear model trained with data available from all users. All document-rating pairs seen by the system so far are used to train the model. This is equivalent to an user independent system modeling approach.

Moving Average (−−) This simplistic model simply returns the average of the ratings seen so far for this user. This is effectively equal to the simplest personalization: only modeling a user's rating bias.

We also include a simplistic baseline measure which corresponds to rating every document as 5. In both user studies, users only provide feedback on documents they viewed/clicked and so the baseline measure loosely corresponds to interpreting click-throughs, the most common type of implicit feedback, as positive feedback.

For each model, at every time step, we evaluate it on the next document in the sequence [3]. Since the objective of the system is to minimize the squared error (L_2 loss), we use it as our evaluation measure.

6. EXPERIMENTAL RESULTS

Table 2 reports the macro-average performance of each model. In all cases the hierarchical model's (Prior) improvement over the user-independent model (No Prior) and non-personalized model (Generic) is statistically significant. For all data sets from the Zhang study it also performs significantly better than the Moving Average. Note that in both cases where only implicit feedback is used, Z_I and C, the Moving Average performs better than the No Prior and Generic models.

Since a major focus of our study is the "cold start" problem, we did some further analysis to study the performance as a function of time or, equivalently, the number of training examples. The error of each model's performance on the Z_{ID} data set is in Figure 2. The performance of each model at each time point is the average of all users, and the performance of a user at a time point is estimated using the average over a 50 document window; this gives us a smoother picture of the general trends in performance. The number of labeled documents from each user varies. If a user's number of labeled data is smaller than the number indexed by the x-axis, the user performance can not be estimated and not included in the average on that time point. Thus the number of users is 15 at the very beginning when the number of training examples is small, and the plot stops when there are fewer than 5 users left for evaluation.

[3]If the model's prediction of whether the user likes a document or not is higher than 5 or smaller than 1, the value is set to 5 or 1. This clearly violates some of the assumptions underlying our model, but it is an simple step that anyone developing such a system would take.

	Zhang			Claypool
	Z_D	Z_{ID}	Z_I	C
Prior	**0.887**	**0.880**	**1.017**	**1.440**
No Prior	1.021	1.118	1.141	2.762
Generic	1.027	1.012	1.157	1.648
Moving Average	1.037			**1.446**
Baseline	3.191			4.692

Table 2: Performance of different models averaged over time and user (in that order). For all datasets except C the hierarchical model performed better than the remaining methods according to a Wilcoxon signed rank test with 95% confidence. On the Claypool dataset the 0 model did not significantly outperform the Moving Average.

Figure 2 shows that the hierarchical user model outperforms the No Prior model and the Generic user model for the first 100 documents. Compared to No Prior, the use of a prior learned from other users helps the initial performance. The figure also shows that the prior and non-prior models converge as more data from a user is available. These observations are what we expected. We truncate the x-axis to 100 to see more detail in the performance in the earlier stage of learning (Figure 3). Here the performance of a user at a time point is estimated using the average over a 10 document window.

Figures 2, 3 show the Generic model performs very similarly to the hierarchical model for the first 200 documents. This suggests that the effect of 'personalization' does not occur until quite a bit of user feedback is gathered. In both studies, each user is interested in several topics (as compared to TREC style single topic user profile) and, therefore, there is a necessary increase in model complexity. As a result, a significant amount of data is needed for effective personalization.

The simple Moving Average model performs surprisingly well, significantly better than the baseline and among the best at the early stage of learning. This demonstrates the importance of user bias in rating, which is the only thing the moving-average tries to model. The Moving Average performance seems to degrade over time, indicating some kind of shift in how the users are rating the document. This may happen if the user suddenly finds a good news source or if the user has a new interest. User bias in rating has been previously observed and explicitly modeled [14]. Our results not only confirm the prior work but also suggest a need for dynamic modeling of user-bias.

Figure 4 shows each model's performance on the Z_I data set. Compared to Figure 2, the prediction power of each model using only implicit feedback is much worse than using all information. The hierarchical model is not significantly different than the Moving Average when the number of training data is smaller than 200, while No Prior performs even worse than the Moving Average. Figure 4 also shows that the Prior and No Prior models are better than

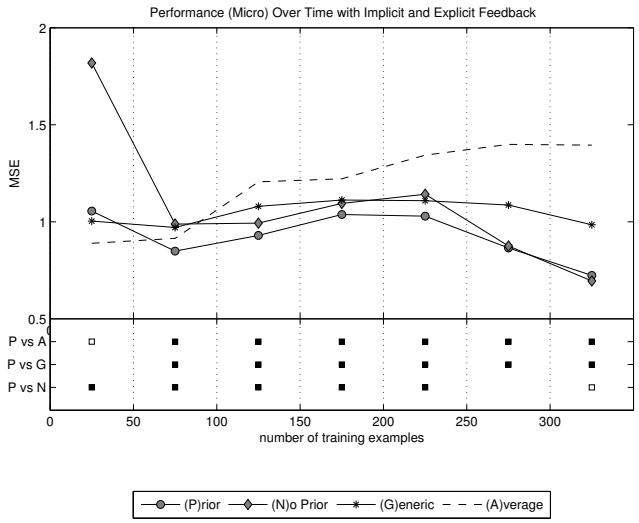

Figure 2: Mean squared error for Z_{ID} of model with Prior, with No prior, Generic user-independent model, and the Moving Average estimator. The bottom part of plot shows the results of a 95% confidence t-test comparing the Prior model to the others. A solid block indicates that the Prior performed better, an empty block indicates it performed worse, and no block means the difference in performance was statistically insignificant.

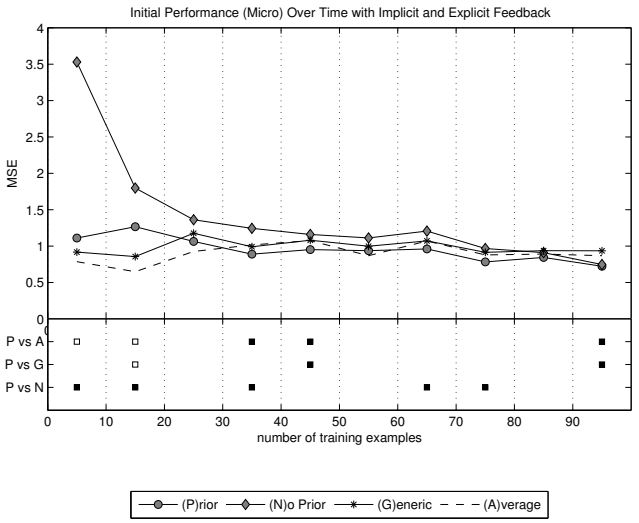

Figure 3: Mean squared error for the first 100 documents in Z_{ID}. See Figure 2 caption.

the Moving Average when sufficient training examples (more than 200) are available. This suggests that a large amount of implicit feedback from a particular user could be useful, which is consistent with Teevan, et al.[15].

Results for Claypool's data set (C) are shown in Figure 5. The range of the x-axis is much smaller than that of Figure 2, because the number of examples per user was far fewer in this study. The hierarchical user model performs similar to the moving-average, while the non prior model performs worse than. This might due to the fact that the system has not accumulated enough implicit feedback to make them useful. Thus, the relative performance of the models is similar to that of the early stage of Z_I, which also only contains implicit feedback. Compared with Z_I, the absolute performance on this data set is worse. The task of Claypool's user study was unrestricted, thus the correlation between implicit feedback and user rate is smaller than Z_I. This is not surprising since previous studies on the usefulness of implicit feedback have reported both negative and positive results [9][13]. In particular Kelly and Belkin showed there to be very little correlation across tasks between time spent viewing a page and how useful they thought the page was. Our results support Kelly's conclusions despite the fact that over the entire data set C, with all user's aggregated, there is a slightly positive correlation between the time spent viewing a page and user rating.

Finally, in order to compare the performance across different feature sets we give a side-by-side plot of the hierarchical model performance on the data sets Z_I, Z_{ID}, and Z_D (Figure 6). It shows that the implicit feedback is unreliable by itself and has only marginal value when combined with explicit feedback.

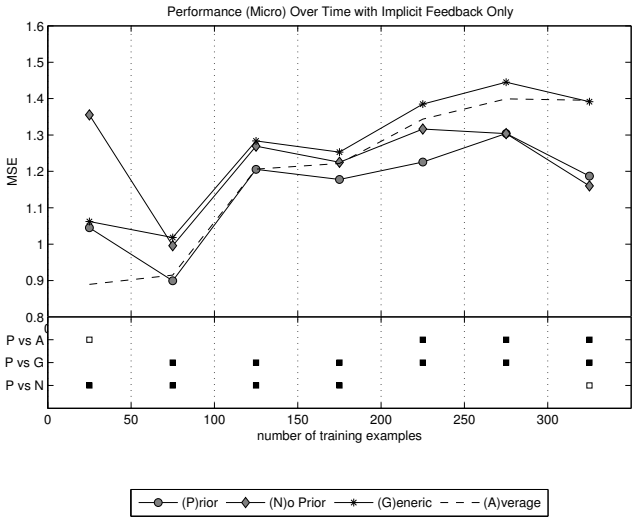

Figure 4: Results for Z_I. See Figure 2 caption.

6.1 Further Discussion

In order to better understand why implicit feedback does not help at the early stage of learning and becomes useful at the later stage when there is a large amount of data, we can decompose the generalization error of a learning algorithm into 3 parts:

- Bias: Measure of how closely the learning algorithm is able to approximate the best solution.
- Variance: Measure of how sensitive the learning algorithm is to the training sample.
- Noise: Measure of the inherent irreducible uncertainty of the problem. For example, for a given x there are more than one possible y.

In general, one wants to use a learning algorithm with both low bias and low variance. However, there is a natural "bias-variance trade-off" for any learning algorithm[5].

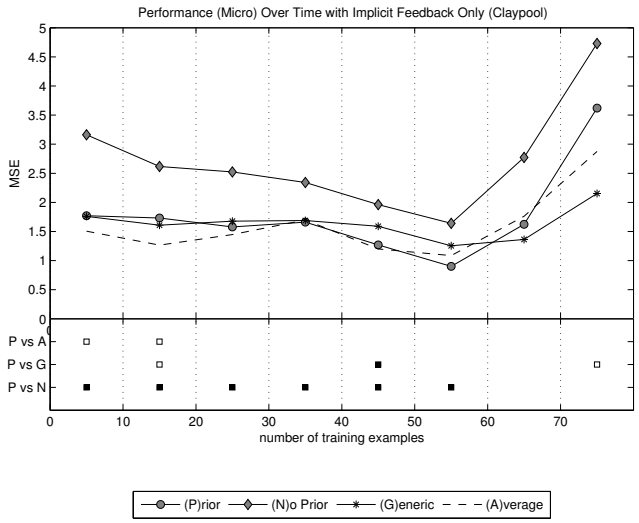

Figure 5: Results for C. See Figure 2 caption.

Bias-Variance Dilemma As the complexity of the learning algorithm increases, the bias goes down, but the variance increases.

The purpose of user profile learning is to find a predictor with a good balance between bias and variance. Adding implicit feedback increases the complexity of the learning algorithm, reducing the bias and increasing the variance. When there are very few training samples, the variance may be the dominant contributor to the generalization error, thus a less complex learning algorithm with less variance such as Moving Average is preferred to a more complex learning algorithm such as No Prior. When there are many training samples, the bias can be the dominant contributor to the generalization error, thus No-Prior is preferred. One needs to be very careful while using implicit feedback. Although informative, implicit feedback may hurt the performance when the amount of training data is small and the learning algorithm is not well regularized. The hierarchical Bayesian modeling approach uses the prior learned from other users as a regularizer to control the model complexity and balances bias and variance based on the amount of training data available. Thus, it consistently works well.

This work is motivated by the fact that there is substantial variation in behavior across users; an example of this is a user's mouse-keyboard preference. It is illustrative to look at the behavior of a 'different' user's model over time. The average correlation coefficient between time-spent-on-mouse and the rating given to a document across users is 0.01. The correlation coefficient between these two variables for user 15 is 0.18, the highest among all users . The weight associated with this feature for user 15, over time, is illustrated in Figure 7. Both the hierarchical and non-hierarchical model demonstrate an adaptive behavior; the generic model does not change much at all. But observe that the hierarchical model is far more stable in the early stages of training. In the later stages it seems to have converged with the non-hierarchical model.

All models used in this paper implicitly assume that user behavior is consistent over time. In reality this is hardly ever maintained. The fact that the performance of all mod-

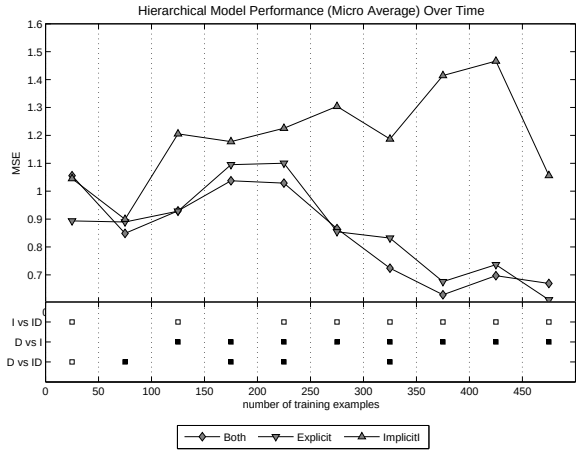

Figure 6: Squared error of hierarchical model using different feature sets. Z_I includes only implicit indicators, Z_D only includes document content features, and Z_{ID} includes both.

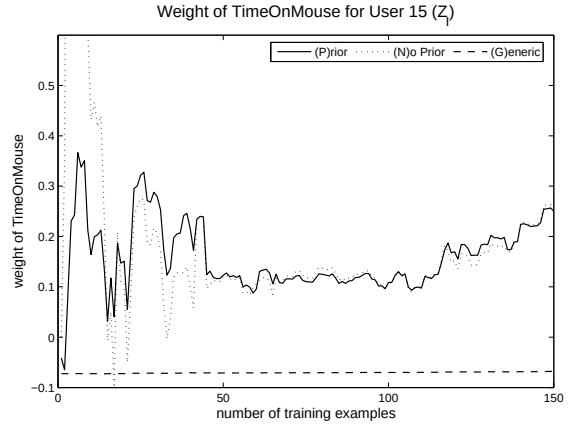

Figure 7: Weight of time-on-mouse, over time, for user 15.

els does not increase monotonically as the number of training data increases (Figure 2, 4, 5) seems to suggest shifting user behavior. And, in fact, we observed dramatic shifts, over time, in the correlation between certain features and document rating. How to model these shifts is an interesting, and important, topic for future work in personalization.

We used linear model to represent a user model, because 1) the earlier work on text classification shows that a well regularized linear model works very well for text classification compared to other state of art linear models such as SVM or logistic regression [11]; 2) it is computationally efficient. The drawback to our approach is that it assumes the relationship between various information about the document and the user rating is well approximated with a linear function. The assumption may be wrong, and the linear model is far from the optimal modeling approach for our task. A first step towards increasing performance would be to try an alternative set of models, such as regression trees, or mapping the features to a non-linear space.

It is worth mentioning that our simulation of the online learning is different from the real scenario, and only documents a user clicked and evaluated are used for training and testing in our experiments. The significant improvement of our approach over the baseline demonstrates that all learning approaches are better than treating click through as positive feedback. Further investigation into the documents not clicked is a critical step to fill the gap between the paper and a real system.

7. CONCLUSION

Variation in how users interact with documents motivates the need for user specific modeling (personalization). A personalized system needs to work well for a new user although the information about the user is limited. This paper explores the usefulness of cheap implicit feedback and borrowing information from other users to improve early performance. We use hierarchical Bayesian models as a unified framework to achieve the two goals. Based on the conditional independence relationship implied by the hierarchical model, we use an efficient training technique to learn a user specific model over time.

On two data sets collected from user studies carried out by two different research groups, we demonstrate that the hierarchical Bayesian modeling approach effectively trades off between information from other users (prior) and user-specific information (data) based on the number of training data from the current user, alleviating poor initial performance for each user. Second, we evaluate the benefits of utilizing implicit feedback in user profiling and find that it is only marginally useful, even when combined with explicit feedback. Compared to the baseline of treating every clicked document as relevant, all learning techniques studied in paper perform significantly better.

The personalization problem is far from solved and our work is just one step in this direction. The work described here poses more questions and challenges in building a personalized adaptive retrieval/filtering system. Our results suggest that the performance of a system is influenced by some other hidden factors, such as the shifting of user behavior or feature noise, both of which could overwhelm the benefit of the modeling approach.

8. ACKNOWLEDGMENTS

This research was funded in part by Google Research Award. Any opinions, findings, conclusions or recommendations expressed in this paper are the authors', and do not necessarily reflect those of the sponsors.

9. REFERENCES

[1] J. Callan, A. Smeaton, M. Beaulieu, P. Borlund, P. Brusilovsky, M. Chalmers, C. Lynch, J. Riedl, B. Smyth, U. Straccia, and E. Toms. Personalisation and recommender systems in digital libraries. Technical report, Joint NSF-EU DELOS Working Group, 2003.

[2] H. Chipman, E. George, and R. McCulloch. Bayesian cart model search (with discussion). *Journal of the American Statistical Association*, 3:935–960, 1998.

[3] M. Claypool, P. Le, M. Wased, and D. Brown. Implicit interest indicators. In *IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces*, pages 33–40, New York, NY, USA, 2001. ACM Press.

[4] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Trans. Inf. Syst.*, 23(2):147–168, 2005.

[5] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Comput.*, 4(1):1–58, 1992.

[6] J. Goecks and J. Shavlik. Learning users' interests by unobtrusively observing their normal behavior. In *IUI '00: Proceedings of the 5th international conference on Intelligent user interfaces*, pages 129–132, New York, NY, USA, 2000. ACM Press.

[7] F. Hastie, Tibshirani. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[8] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2002.

[9] D. Kelly and N. J. Belkin. Display time as implicit feedback: understanding task effects. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 377–384, New York, NY, USA, 2004. ACM Press.

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

[11] F. Li and Y. Yang. A loss function analysis for classification methods in text categorization. In T. Fawcett and N. Mishra, editors, *ICML*, pages 472–479. AAAI Press, 2003.

[12] D. Nichols. Implicit rating and filtering. In *Proceedings of 5th DELOS Workshop on Filtering and Collaborative Filtering*, pages 31–36. ERCIM, 1998.

[13] B. Pan, H. A. Hembrooke, G. K. Gay, L. A. Granka, M. K. Feusner, and J. K. Newman. The determinants of web page viewing behavior: an eye-tracking study. In *ETRA'2004: Proceedings of the Eye tracking research & applications symposium*, pages 147–154, New York, NY, USA, 2004. ACM Press.

[14] L. Si and R. Jin. A flexible mixture model for collaborative filtering, 2003.

[15] J. Teevan, S. T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th Annual ACM Conference on Research and Development in Information Retrieval*, 2005.

[16] R. W. White, J. M. Jose, and I. Ruthven. An implicit feedback approach for interactive information retrieval. *Information Processing and Management*, 42(1):166–190, January 2006.

[17] K. Yu, V. Tresp, and S. 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, New York, NY, USA, 2004. ACM Press.

[18] 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.