Active Collaborative Prediction with Maximum Margin Matrix Factorization

Irina Rish and Gerald Tesauro
IBM T.J. Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532
rish,gtesauro@us.ibm.com

Abstract

Collaborative prediction (CP) is a problem of predicting unobserved entries in sparsely observed matrices, e.g. product ratings by different users in online recommender systems. However, the quality of prediction may be quite sensitive to the choice of available samples, which motivates active sampling approaches. In this paper, we suggest an active sampling method based on the recently proposed Maximum-Margin Matrix Factorization (MMMF) (Srebro, Rennie, & Jaakkola 2004), a linear factor model that was shown to outperform state-of-art collaborative prediction techniques. MMMF is formulated as a semi-definite program (SDP) that finds a low-norm (rather than traditional low-rank) matrix factorization, and is also closely related to learning max-margin linear discriminants (SVMs). This relation to SVMs inspires several margin-based active sampling heuristics that augment MMMF and demonstrate promising results in a variety of practical domains, including both traditional recommender systems and novel systems-management applications such as predicting latency and bandwidth in computer networks.

1 Introduction

Given a large but sparsely sampled matrix, the *collaborative prediction (CP)* problem is to predict the unobserved entries from the observed samples, assuming the entries are dependent. Typical application include online recommendation systems that attempt to predict user's preferences towards different products (e.g., movies, books), based on previously obtained product ratings from different users. Collaborative prediction can be also applied to non-traditional domains such as distributed systems management applications considered in this paper. In such applications, we wish to predict the end-to-end performance, such as connectivity and/or latency in computer networks or bandwidth in peer-to-peer content-distribution systems, based on a limited number of available measurements between pairs of nodes. Moreover, collaborative prediction tasks may arise in various other domains, e.g. in image processing, where we may want to reconstruct unobserved (occluded) parts of an image from the observed pieces.

A typical assumption that leads to various collaborative prediction techniques is a *factorial* model that assumes the presence of some hidden factors that affect user's preferences towards the products. For example, genre of a movie, its comic factor, and its violence factors may affect user's preferences. Similarly, two nodes that are located in same part of the network may share several "hidden factors" such as intermediate nodes on their path to a third node; moreover, even distant nodes can share some other hidden factors which determine a quality of service they provide: e.g., a high-bandwidth can be achieved by downloading from a remote but powerful server instead of local laptop with a wireless connection. In this paper, we will focus on *linear factor models* which result into a matrix-factorization approach to collaborative prediction.

The predictive accuracy of such models can improve dramatically when more samples become available; however, sampling can be costly: a user may become annoyed if she is asked to rate many products or a network may become congested if too many measurements are performed. Besides, suggesting a product to buy or a server to download from has a high cost if the user does not like the product, or the download bandwidth turns out to be low. Therefore, a cost-efficient active sampling becomes an important component of any successful collaborative prediction approach.

In this paper, we propose an active-learning extension of the recently proposed Maximum Margin Matrix Factorization (MMMF) approach to collaborative prediction that was shown to outperform state-of-art collaborative prediction methods and has some nice theoretical guarantees (Srebro, Rennie, & Jaakkola 2004; Rennie & Srebro 2005). MMMF is a matrix factorization approach formulated as a convex optimization problem that uses low-norm constraints, unlike previous non-convex approaches, such as low-rank (SVD-like) or non-negative matrix factorizations (Lee & Seung 2000). Besides, MMMF is closely related to maximum-margin linear discriminants (SVMs), i.e. it can be viewed as simultaneous learning of multiple SVMs and a set of features common to to all SVMs. This insight is directly exploited by our active learning approach that extends MMMF with margin-based active-learning heuristics, where the margin is used to estimate informativeness of a candidate sample, as suggested in (Tong & Koller 2000). Besides the straightforward "most-uncertain" (min-margin) sample selection, we also investigate alternatives that take into account the cost of sampling.

Previous work on active sampling for collaborative filtering includes a *value-of-information* approach of (Boutilier, Zemel, & Marlin 2003) and Bayesian model averaging method of (Jin & Si 2004). Both approaches are based on probabilistic hidden-factor models and computationally expensive procedures for choosing next active sample that require minimization of expected cost (or uncertainty). On the contrary, our active sampling is quite simple and inexpensive as it only compares the margin values produced by MMMF. Another related work proposes an active-sampling method for low-rank matrix factorizations (Drineas, Kerenidis, & Raghavan 2002) that requires a small number of users to provide the ratings of ALL products – a clearly unrealistic assumption in any large enough, practical recommendation system. Although our heuristic active sampling lacks theoretical guarantees associated with the above approach, it is much more practical since it does not impose any unrealistic sampling assumptions. Empirical evaluation on several application domains, from recommender systems to computer networks and peer-to-peer files distribution systems, demonstrate the advantages of our active sampling methods.

In summary, this paper makes following contributions. It proposes a simple, computationally efficient active sampling extension of the state-of-art MMMMF method for collaborative prediction, compares several active-sampling strategies, both on traditional collaborative filtering domain (movie rating prediction) and on novel application domain – distributed computer systems management, and demonstrates a noticeable improvement in prediction accuracy over random sampling.

2 Collaborative Prediction as Matrix Factorization

Collaborative prediction problem can be stated as follows. Given a partially observed $n \times m$ matrix Y, let us find a matrix X of the same size that provides "best" approximation for unobserved entries of Y with respect to a particular *loss function*, such as sum-squared loss for real-valued matrices, 0/1 loss or its surrogates such as hinge loss for binary and ordinal matrices, and so on.

Linear factor models, a particular type of factor models for collaborative prediction, assume that each factor is a preference vector, and actual user's preferences correspond to a weighted linear combination of these factor vectors with user-specific weights. Let k be the number of such factors, then the matrix Y can be approximated by a *matrix factorization* $X = UV$, where U is a $n \times k$ *coefficient matrix* (where each row represents the extent to which each factor is used) and V is a $k \times m$ *factor matrix* where the rows represent the "expression level" of the factors in each of m "products". Since the rank of the approximation matrix X is clearly bounded by k, fixing k to some small value leads to a low-rank matrix factorization approaches.

For example, a standard matrix-factorization approach is singular value decomposition (SVD) which finds a low-rank approximation that minimizes the sum-squared distance between X and a *fully observed* Y. The problem is, when Y is not fully observed, as in collaborative prediction and particularly in end-to-end performance prediction, SVD is not directly applicable and finding a low-rank approximation to a partially observed function using a sum-squared loss becomes a difficult non-convex optimization problem, for which no exact solution method is known. Also, even for completely known matrix Y, approximating it with respect to other losses that the sum-squared loss (e.g., expected classification error) is still a non-convex optimization problem with multiple local minima (Srebro, Rennie, & Jaakkola 2004).

In order to overcome such limitations, a novel *Maximum Margin Matrix Factorization (MMMF)* approach was proposed by (Srebro, Rennie, & Jaakkola 2004). This approach replaces the bounded-rank with the *bounded norm* constraint on U and V and yields a convex optimization problem. Namely, Lemma 1 in (Srebro, Rennie, & Jaakkola 2004) shows that finding the matrices U and V having low Frobenius norms $\|U\|_{Fro}$ and $\|V\|_{Fro}$ is equivalent to minimizing the *trace-norm* (the sum of singular values) $\|X\|_{\sum}$ of X, since

$$\|X\|_{\sum} = \min_{X=UV} \|U\|_{Fro}\|V\|_{Fro} = \min_{X=UV} \frac{1}{2}(\|U\|^2_{Fro}+\|V\|^2_{Fro}) \quad (1)$$

Since the trace-norm is a convex function (Srebro, Rennie, & Jaakkola 2004), minimizing it together with any convex loss function or constraint results into a convex problem.

For simplicity, we focus herein on binary-valued matrices $Y \in \{-1, 1\}^{n\times m}$, and thus use the MMMF with hinge-loss, as in max-margin linear discriminant (SVM) learning. The MMMF optimization problem can be then stated as:

$$\min_X \|X\|_{\sum} + c \sum_{ij\in S} h(Y_{ij}X_{ij}), \quad (2)$$

where c is a trade-off constant and $h(z) = \max(0, 1 - z)$ is the hinge-loss, minimizing which is equivalent to minimizing slack variables $\xi_i j \geq 0$ in soft-margin constraints $Y_{ij}X_{ij} \geq 1 - \xi_{ij}$.

Matrix factorization can be also viewed as a simultaneous learning of feature vectors and linear classifiers. Assume a factorization $X = UV$ is found, the rows of the $n \times k$ matrix U can be viewed as a set of n *feature vectors*, while the columns of V can be viewed as *linear classifiers*, and the entries of the matrix X are the results of classification using these classifiers. The original entries in the matrix Y can be viewed as *labels* for the corresponding feature vectors, and the matrix factorization task can be interpreted as finding simultaneously a collection of feature vectors (rows in U) and a set of linear classifiers (columns in V), given a set of labeled samples (columns in the original matrix Y), such that a good prediction of unobserved entries can be made. Particulary, the MMMF formulation above can be viewed as learning a collection of maximum-margin classifiers (SVMs) simultaneously with learning a common set of features.

3 Active Learning with MMMF

Standard collaborative prediction approaches, including MMMF, assumed no control over the data collection process. However, we have a choice between different actions

that provide us with new samples. For example, in online recommendation systems, we choose a product suggested to the current user; in network latency prediction, we can request a probe (e.g., ping) between a particular pair of nodes; in content distribution systems, we can suggest a mirror site for a file download, and so on. Such additional measurements can greatly improve the predictive accuracy of our model, but they also have a cost (e.g., potentially low bandwidth or high network latency if a server is not selected carefully). One one hand, we wish to choose the next sample which is most-informative and leads to greatest improvement in the predictive accuracy in the future (i.e., yields better exploration), while on the other hand we want to avoid choosing samples which might be too costly by exploiting our current predictions about the sample costs (i.e., the corresponding predicted performance). Such exploration vs exploitation trade-offs must be considered as a part of our decision-making.

As mentioned in the previous section, MMMF approach can be viewed as learning a collection of SVMs, which provides a natural way for combining MMMF with various *active learning* approaches developed for SVMs. In this paper, we a simple heuristic margin-based approach, that uses the margin as our confidence estimate in the predictions made, similarly to active learning approach of (Tong & Koller 2000). Namely, (Tong & Koller 2000) suggest to choose next the the *minimum-margin* sample, i.e. the one which is closest to the separating hyperplane, and can be viewed as the one we are least confident about. This heuristic was shown to be successful in practice, and is very efficient computationally[1]. The idea of min-margin active sampling is demonstrated in Figure 1.

Besides the "aggressive" *most-uncertain* sampling we also tried several other active sampling approaches that take into account the cost of sampling and may decide to be more "conservative" about sample choice, e.g., when sampling also means providing a service such as file download, where besides improving the future accuracy we are also concerned with the immediate cost of sampling. We assume binary prediction problems (e.g., the performance over or under a specified threshold) and assume that positive samples (e.g., high bandwidth or product ratings) have less cost than the negative samples. We then explore several "cost-conscious" active learning heuristics, such as *most-uncertain-positive* heuristic that chooses positive min-margin sample, as well as *least-uncertain* (max-margin) and *least-uncertain-positive* heuristics, which which should corresponds to prediction we are most confident about. However, such sample selection may lead to a less accurate model, as we show in the empirical section where the different sampling heuristics are compared on several data sets.

Our active sampling algorithm (Active MMMF, or A-MMMF) is presented in figure 2. The algorithm assumes

[1]Although min-margin heuristic may be ineffective for problems with large label noise close to the separating hyperplane, as noticed by (Bordes *et al.* 2005), in many collaborative prediction settings there is little or no noise in labeling: e.g., user's preferences for a movie typically do not change.

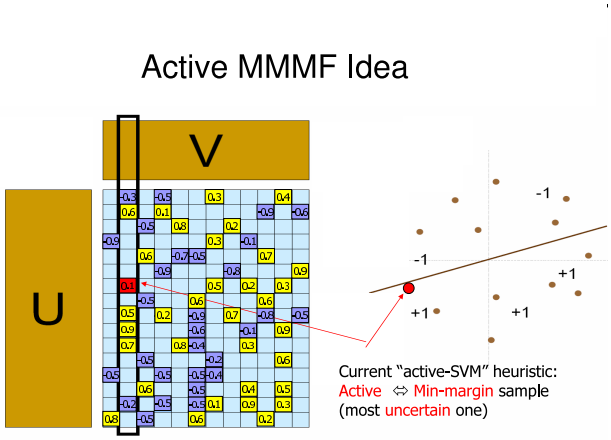


Figure 1: Main idea of active learning in MMMF: choose the most "uncertain" sample next, where the distance to linear classifier (which we will also call "margin" here) measures the confidence in the prediction (i.e. we are least confident in predictions made for the instances closest to separating line between positive and negative examples). Note that the matrix is real-valued, with X_{ij}=distance (with sign) between feature-vector i and linear classifier j.

a particular active learning heuristic specified as an input.

4 Empirical Evaluation

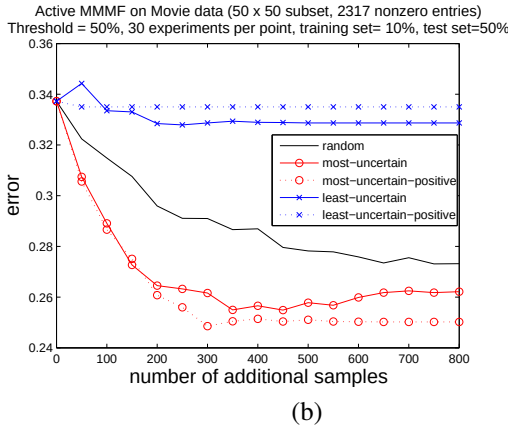
We tested active learning approaches described above on the data from various practical applications. We select a subset of most populated rows and columns, to increase matrix density for testing purposes. We then split each dataset into a training, testing and active subsets, where active subset simulates the source of active samples. A training set is typically selected to be quite small (e.g., 5% of the whole dataset), to imitate learning "almost from scratch". We plot the prediction error on the training dataset, for each of the active strategies compared random sampling of the same number of instances.

The first dataset, called **Movies**, includes movie ratings collected through user interactions with the site www.movielens.org. This includes ratings on the scale of 1 (worst) to 5 (best) by 500 users of 1000 movies. We selected a subset of 50 users and 50 movies that correspond to most-populated rows and columns. We then impose a threshold to make the data binary, i.e. we assume that the rating larger than 3 is considered "good". The results are presented in Figure 3a. We can see that the most-uncertain sampling provides a significant improvement over the random sampling, while the max-margin sampling, as expected, is not very informative and practically does not improve the error. We also computed the actual cost of sampling, assuming no cost for positive samples selected and unit cost of the negative ones, and plotted it in Figure 3b. Clearly, random sampling would roughly have the slope of the cost curve equal to the proportion of negative samples in the data. Surprisingly, the alternative strategies did not deviate significantly from this random-sampling linear cost growth, although we can see some deviation for larger number of samples. We can see

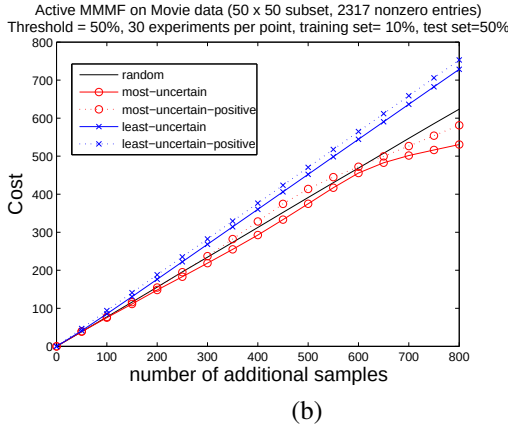
Active Max-Margin Matrix Factorization (A-MMMF)

Input: Sparse binary (-1/1) matrix $\mathbf{Y}$, batch size k, max # of new samples N, active-sampling heuristic h (most-uncertain etc).
Output: Full binary-valued matrix $\mathbf{X}$ predicting unobserved entries of $\mathbf{Y}$.
Initialize: Y' = Y /* currently observed data */
$N' = 0$ /* current number of active samples */
1. $\mathbf{X}'$ = MMMF(Y') /* compute full real-valued matrix X', where $|X'_{ij}|$=distance(feature-vector i, linear classifier j), $sign(X'_{ij})$ predicts unseen Y_{ij}. */
2. U = set of unobserved entries of Y'
3. $S = active\_select(h, X', U, k)$ /* select k best unobserved samples from U using heuristic h and current predictions X' */
4. Request labels for new samples $s_{ij} \in S$, and add them to Y'.
5. If $N' + k < N$
$N' = N' + k$; goto 1
else return $sign(X')$ /* return only binary -1/1 predictions */

Figure 2: Active max-margin matrix factorization (A-MMMF) algorithm.



(b)



(b)

Figure 3: Prediction results on Movie dataset: (a) prediction accuracy and (b) total cost of sampling.

that the most-uncertain and most-uncertain-positive strategies are actually better not just in terms of future predictive error, but also in terms of total sampling cost.

Similar results were observed in multiple systems management domains. These include several publicly available network latency datasets that we borrowed from (Mao & Saul 2004) and proprietary data from an IBM-internal file distribution system, described in more detail below.

PL-RTT2003 dataset (from (Mao & Saul 2004)) was originally obtained from PlanetLab pairwise ping round-trip time (RTT) measurement project (Stribling). A subset of *minimum* round-trip times measured at 3/23/2004 0:00 EST was selected, and missing values were filtered out, since none of the algorithms used in (Mao & Saul 2004) could handle missing values. Namely, if there where some missing values either in a column or a row corresponding to some node, both the row and the column corresponding to such node were eliminated from the matrix (Mao 2006).

PL2005 dataset was obtained directly from the PlanetLab project, and *average* round-trip times measured at 02/01/2005 0:00 were selected. We eliminated only rows or columns that corresponded to completely missing measurements (e.g., node serving only as a source or only as a destination), but unlike (Mao & Saul 2004) we did not eliminate the corresponding node from the matrix, so the resulting matrices are not necessarily square.

NLANR-AMP dataset was obtained by (Mao & Saul 2004) from the NLANR Active Measurement Project (NLA), that collects measurements between the pairs of nodes at NSF supported HPC cites, with about 10% of the nodes located outside of the US. All-pairs measurements were collected over 110 nodes on 01/30/2003; each host was pinged once per minute, and the minimum response time per day was chosen for each pair of nodes.

P2PSim dataset was obtained from the P2PSim project (P2P) that measured network latency among 2000 Internet DNS servers using King method (Gummadi, Saroiu, & Gribble) (the servers were taken from the Internet-scale Gnutella network trace).

dGrid2005 dataset was collected from *downloadGrid*, an IBM-internal file distribution system; the data were collected in Dec. 2005 over 10913 clients and 2746 servers, and contain the history of file downloads. The architecture of downloadGrid is similar in some respects to the Internet-based Gnutella, Napster and BitTorrent protocols as it allows peer-to-peer file downloading. However, it differs in utilizing a centralized decision-making architecture for matching "servers" (i.e. sources for downloadable files) with "clients" (i.e. download destinations). While the use of centralized decision-making is motivated primarily by security issues, the concomitant centralized data collection provides an opportunity for optimization of global system performance. By running algorithms such as MMMF on the aggregate system data, it may be possible to make reasonably accurate performance predictions for unobserved client-server pairs, and thereby make better assignments than could be made based solely on directly observed performance data.

From the history of file downloads in the downloadGrid, we created a matrix where each entry corresponds to the av-

erage bandwidth for a given (client, server) pair. The original matrix was extremely sparse (only less than 0.3 % of the entries were observed) in that recorded performance data for any given node contains interactions with only a few other nodes. In order to have enough data for testing the learned model, we have selected a dense submatrix by choosing 70x70 subset of the clients and servers that yield rows and columns most densely populated with recorded performance data. Currently, we also imposed same 70x70 size restrictions on the other datasets.

All of the real-valued performance measurements in each data set were transformed to a -1/1 binary representation by comparing the measured performance with a given threshold, typically chosen to lie at a certain percentage level (50%, 70%, or 90%) of the entire set of performance statistics in the particular dataset (i.e. 50% corresponds to median performance).

We now present our results applying A-MMMF (MMMF augmented by active sampling) to the datasets described above. Our evaluation methodology is as follows. For each 70x70 dataset, we perform 20 independent trials, where in each trial we initialize the MMMF predictor by training on a randomly selected 5% of the matrix elements. We then randomly select another 50% of the matrix elements to be held out as test data. The remaining 45% of matrix elements serves as a source of samples that may be selected by various "active" heuristics. We progressively retrain MMMF in stages by selecting a batch of 50 samples from the active set, transferring them to the training set, and then recomputing the MMMF solution. We compare five different active selection heuristics here: "Most-uncertain" chooses samples that are closest to the margin of the current MMMF predictor. "Most-uncertain-positive" also uses the min-margin idea but further constrains the selected samples to be estimated positive (i.e. above threshold) by the current MMMF predictor. "Least-uncertain" and "Least-uncertain-positive" apply the opposite principle of choosing samples that are furthest from the margin, i.e., samples for which MMMF is most confident in its prediction accuracy. Finally, "random" denotes the baseline uniform random sampling strategy.

Our results, plotted as mean test-set prediction error vs. number of additional samples selected, are shown in Figures 4. (We did not plot the error-bars here to avoid the clutter, but the differences were statistically significant.) The qualitative behavior seen in each dataset is highly consistent. In each case, we observe as expected that both of the "most-uncertain" strategies reduce prediction error more rapidly than with random sampling, and that both of the "least-uncertain" strategies provide quite poor choices of training samples, leading to very little improvement in prediction accuracy. Additionally, we note that, in accordance with active learning theory, the number of samples needed to reach a desired accuracy level may be significantly reduced when using the "most-uncertain" heuristics compared to using random sampling. For example, if we wish to reach the apparent asymptotic performance levels of the "most-uncertain" strategies, we would need only $\sim$500 active samples in the PL-RTT2003 dataset, $\sim$700 samples in the NLANR-AMP dataset, and $\sim$1000 samples in the P2Psim dataset, vs. more

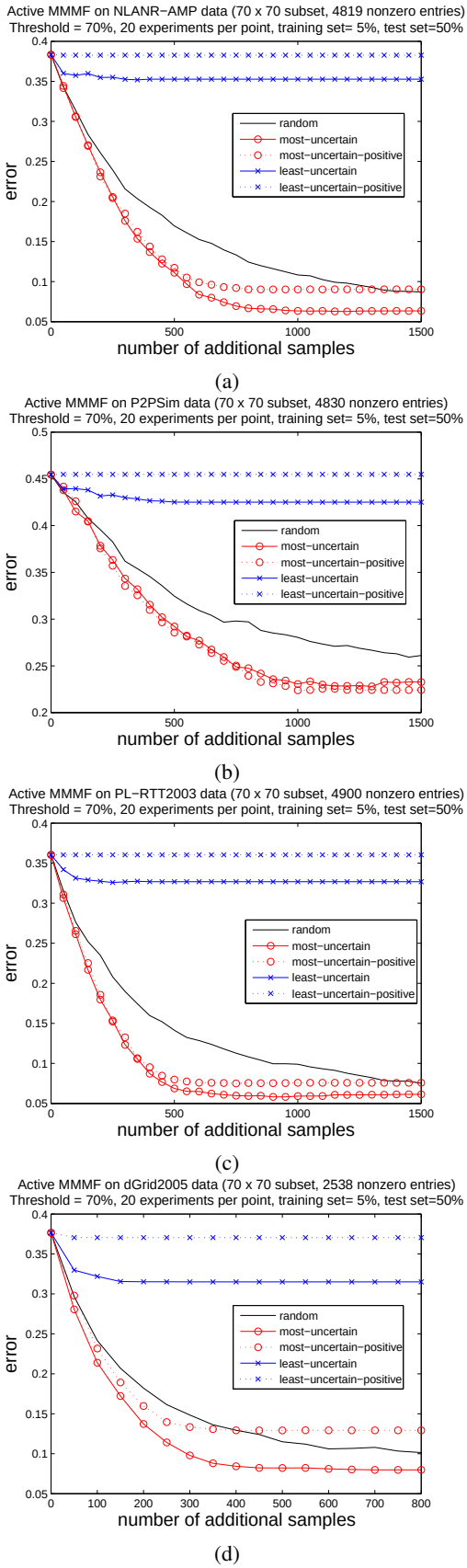


Figure 4: Active learning results on (a) NLANR-AMP, (b) P2PSim, (c) PL-RTT2003 and (d) dGrid2005 datasets: improvement in the prediction error with the increasing number of samples.

than 1500 randomly chosen samples in each of these cases. A final point of interest is that the accuracy of the "most-uncertain-positive" strategy generally lies close to that of the "most-uncertain" strategy, and thus may provide a somewhat safer alternative in scenarios where choosing a sample that turns out to be negative (below threshold) entails a tangible cost of delivering poor performance to a customer.

In addition to minimizing the number of samples needed to train an effective CP module, it is perhaps of more salient interest to minimize whatever costs may be associated with acquiring the training samples. We have also examined this issue within our binary performance model by formulating a sampling cost model that is dominated by the SLA cost of poor performance. In this model, whenever a sampling strategy chooses a "negative" sample with below-threshold performance, we assign it a unit cost, whereas selecting a "positive" sample with above-threshold performance incurs no cost. In this way we can measure for each of our datasets the total sampling cost needed to reach a given prediction accuracy level, using each of our five candidate sampling strategies. Our results for four of our datasets are plotted below in Figure 5. We note in each case that plots of prediction error vs. cumulative cost are qualitatively similar to the corresponding plots of prediction error vs. number of training samples. Perhaps not surprisingly, this suggests a fairly linear relationship between the number of training samples and the total sampling cost, which would occur if negative samples are acquired at a fairly constant rate. We can see that for each dataset, usage of our min-margin sampling heuristics can yield very significant savings in total cost relative to random sampling, if one is interested in reaching low prediction errors where the curves flatten out. In the P2PSim dataset, we can obtain a prediction error of 0.25 at cost of $\sim$270, whereas with random sampling the cost would be over 500. In the NLANR-AMP dataset, we can reach 0.1 prediction error at a cost of $\sim$200 using min-margin sampling, vs. a random sampling cost of $\sim$350. In the PL-RTT2003 dataset, with min-margin sampling we reach 0.1 prediction error at a cost of $\sim$100, vs. a cost of over 300 using random sampling. Finally, in the dGrid2005 dataset, min-margin sampling achieves 0.1 prediction error at a sampling cost of $\sim$100, compared to a random sampling cost of $\sim$200.

5 Conclusions and Future Work

We proposed a simple, computationally efficient active sampling extension of the state-of-art MMMMF method for collaborative prediction and compares several active-sampling strategies, both on traditional collaborative filtering domain (movie rating prediction) and on novel application domain – distributed computer systems management. Promising empirical results are demonstrated on all applications considered.

Our greatest interest in future work is to extend our framework to encompass the dynamic aspects of both end-to-end performance prediction, as well as network management decisions based upon such predictions. Our current formulation ignores the dynamically changing nature of network states, which may render older measurement information

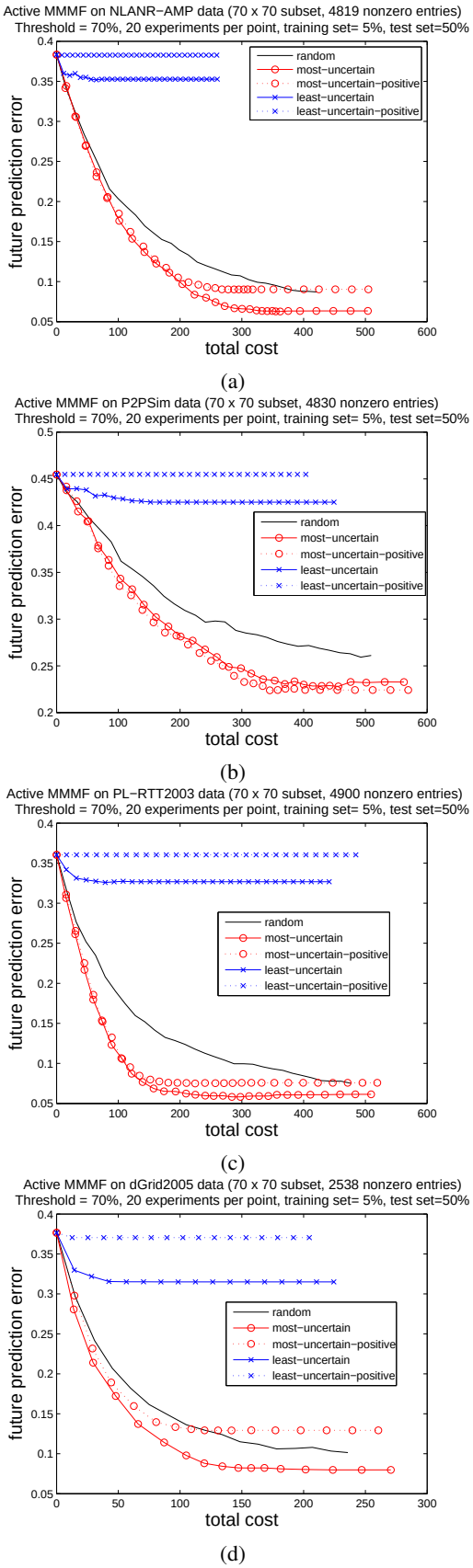


Figure 5: Trade-off between the error reduction and the cost of active sampling on the (a) NLANR-AMP, (b) P2PSim, (c) PL-RTT2003 and (d) dGrid2005 datasets.

obsolete, and the impact of decisions on states, which may necessitate a change in decision-making strategies. For example, if too many clients are directed to download a file from a "good" server, it could cause the server to become a "bad" server due to overloading. One approach we are investigating for dynamic inference is to extend MMMF by adding mechanisms for inference over time based on time-series analysis techniques. This could allow development of models of the decaying influence of older measurements on the current inference matrix, as well as forecasting methods predicting likely future states of the inference matrix based on how it has evolved in the past. The other major extension we are investigating is combining dynamic inference models with models for sequential decision making, e.g., Markov Decision Process models. We are especially interested in Reinforcement Learning approaches to this, which would allow automatic learning of effective management policies without needing explicit models of management actions influence state transitions in the network.

Another important future direction is to further improve the computational efficiency of active MMMF by making it incremental, i.e. reusing the solution obtained on the previous sampling iteration without having to solve the MMMF optimization from scratch.

References

Bordes, A.; Ertekin, S.; Weston, J.; and Bottou, L. 2005. Fast Kernel Classifiers with Online and Active Learning. *Journal of Machine Learning Research* (6):1579–1619.

Boutilier, C.; Zemel, R.; and Marlin, B. 2003. Active collaborative filtering. In *Proc. of UAI*, 98–106.

Drineas, P.; Kerenidis, I.; and Raghavan, P. 2002. Competitive Recommendation Systems. In *Proc. of the 34th ACM Symposium on Theory of Computing (STOC)*, 82–90.

Gummadi, K.; Saroiu, S.; and Gribble, S. D. King: Estimating latency between arbitrary internet end hosts. In *Proc. of the SIGCOM Internet Measurement Workshop (IMW 2002), Nov. 2002*.

Jin, R., and Si, L. 2004. A Bayesian approach toward active learning for collaborative filtering. In *Proc. of UAI*, 278–285.

Lee, D. D., and Seung, H. S. 2000. Algorithms for Non-negative Matrix Factorization. In *NIPS*, 556–562.

Mao, Y., and Saul, L. K. 2004. Modeling Distances in Large-Scale Networks by Matrix Factorization. In *Proc. of Internet Measurement Conference (IMC-04)*.

Mao, Y. 2006. Private communication.

The NLANR active measurement project. http://amp.nlanr.net/active.

The P2PSim Project. http://www.pdos.lcs.mit.edu/p2psim.

Rennie, J., and Srebro, N. 2005. Fast maximum margin matrix factorization for collaborative prediction. In *Proc. of ICML 2005*, 713–719.

Srebro, N.; Rennie, J.; and Jaakkola, T. 2004. Maximum Margin Matrix Factorizations. In *Proc. of NIPS-04*.

Stribling, J. All pairs of ping data for PlanetLab. http://www.pdos.lcs.mit.edu/∼simstrib/pl_app.

Tong, S., and Koller, D. 2000. Support Vector Machine Active Learning with Applications to Text Classification. In *Proc. of ICML 2000*.