Differential Data Analysis for Recommender Systems

Richard Chow*
Intel Corporation
richard.chow@intel.com

Hongxia Jin
Samsung Electronics R&D
hongxia.jin@samsung.com

Bart Knijnenburg*
UC Irvine
bart.k@uci.edu

Gokay Saldamli
Samsung Electronics R&D
gokay.s@samsung.com

ABSTRACT

We present techniques to characterize which data contributes most to the accuracy of a recommendation algorithm. Our main technique is called differential data analysis. The name is inspired by other sorts of differential analysis, such as differential power analysis and differential cryptanalysis, where insight comes through analysis of slightly differing inputs. In differential data analysis we chunk the data and compare results in the presence or absence of each chunk. We apply differential data analysis to two datasets and three different attributes. The first attribute is called user hardship. This is a novel attribute, particularly relevant to location datasets, that indicates how burdensome a data point was to achieve. The second and third attributes are more standard: timestamp and user rating. For user rating, we confirm previous work concerning the increased importance to the recommender of high and low user ratings.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering*

General Terms

Experimentation, Human Factors

Keywords

Recommender Systems

1. INTRODUCTION

Services that use recommender systems have become increasingly common and most of these systems exist by virtue of "big data" stored and used for recommendation and personalization purposes. One example is location recommender systems (e.g., recommending nearby points-of-interest), a large and growing area because of the connection with mobile devices. What part of all this data is really necessary for making good recommendations?

This question is our main motivation for the paper. Our approach starts with techniques to rank data according to usefulness to the recommender system. In essence, we identify attribute values which are associated with usefulness to the recommender. We test the association through what we call *differential data analysis*: we select data according to attribute values and observe the effect of the presence and absence of this data on recommendation accuracy.

Our work has applications to user privacy and data reduction, but due to space considerations we only mention these applications in passing here and defer detailed discussion to the extended version of this paper [1]. One of the main challenges of recommender systems is user privacy, as the data to build a user profile can be potentially sensitive or embarrassing. Adding fake data and suppressing actual data are standard tactics to enhance user privacy, and one consequence of our work is to make these tactics more efficient.

*Work performed while at Samsung Electronics R&D.

RecSys'13, October 12–16, 2013, Hong Kong, China.

http://dx.doi.org/10.1145/2507157.2507190.

2. DIFFERENTIAL DATA ANALYSIS

Our main experimental technique involves dividing the data into chunks and comparing results in the presence or absence of each chunk. More specifically, suppose a data attribute ranks each user's data in some way, for instance by time or user rating. One can then divide the data into chunks based on this ranking. For convenience, we often divide into 10 chunks, or *deciles*. We can then examine the relative effect of a particular decile on recommendation accuracy as follows. We form a training and test set as usual. We rank each user's training data by the data attribute under examination. We remove the first decile of each user's data from the training set and calculate the recommendation accuracy (using a fixed algorithm). We continue by removing the second decile of each user's data, etc., and we end up with 10 readings for the accuracy, which can then be compared. A relatively high accuracy reading implies the corresponding decile is less important to the recommender; a low accuracy reading implies the corresponding decile is more important.

Note that there are other ways to do the differential data comparison. Rather than removing data deciles from all users simultaneously, an alternative approach would consider each user in isolation. Deciles would be removed from one user at a time, the effect on recommendation accuracy

would be measured for that user, and accuracy measurements would be aggregated over all users at the end. For the experiments described in this paper, we used the former approach since it is less computationally intensive.

3. DATA ATTRIBUTES

We study three data attributes in this work. The first attribute we examine is a "user hardship" for each data point, the effort it takes the user to attain the data point. The idea is that data points which are more or less difficult to achieve may be more or less indicative of a user's preferences. This attribute is most intuitively associated with location datasets. To compute the user hardship on our location dataset, a user's location traces can be clustered according to surface-of-the-earth distance, and points are ranked according to their distance to the set of cluster centroids. We call this the KMeans user hardship measure. Points furthest away from the cluster centroids have a higher user hardship score, as these locations are further from a user's usual haunts and require more work to get to. Another way to compute the user hardship is to measure the distance to any other point (rather than the cluster centroid). We call this the Density user hardship measure. We present experimental results in Section 4.1 on user hardship for a location dataset.

The second attribute we investigate is the timestamp of the data, for instance the time of a location checkin or the time a rating is given. We investigate whether certain time periods of the day might correspond to more important data. In Sections 4.1 and 4.3 we present our results on filtering by timestamps for a location dataset and a rating dataset.

Finally, our third attribute has been discussed in previous work: the actual user rating, for instance the number of stars given by a user for a product. Previous work has recognized that data with high and low ratings are more important to the user [5] and to the recommender [3].

4. EXPERIMENTAL RESULTS

We experimented with our techniques using two actual datasets, a location dataset and a movie rating dataset. To measure the effect of various filters on recommendation accuracy, we adopt an application-specific definition of recommendation accuracy, as in [2]. For each dataset, we use standard recommender algorithms and measure the effect of the filters on the output of these algorithms. For example, for movie ratings prediction, we measure the RMSE with and without the filter for the same algorithm.

4.1 Location Dataset

We modeled a location recommender service with a dataset of Gowalla check-ins collected from June to October 2010[1]. We studied mainly Austin, Texas, which was Gowalla headquarters and had the highest amount of activity. We also studied three other cities: Los Angeles, New York, and Dallas (see Table 1).

We considered a simple scheme of each user giving a binary positive rating to each location visited. Locations not visited were considered unrated. We ignored the number of visits. In fact, in our dataset, over 80% of the ratings were the result of a single visit by a user, i.e. checking in multiple times to the same location was less common than checking in once. To give users a realistic chance of checking into any location, we confined ourselves to one city at a time.

[1]We thank Betim Berjani and Thorsten Strufe for this data.

City	# Users	# Ratings	# Locations
Austin	4871	245153	9577
Dallas	2991	103569	5473
Los Angeles	2957	61891	10151
New York	3213	106125	16861

Table 1: Gowalla statistics for four cities studied.

We randomly chose 20% of each user's ratings as a held-out test set. We trained a Top-N recommender based on the remaining ratings. The recommender's task is to predict the likelihood of the remaining (user, location)-pairs.

We used the following standard algorithm for computing recommendations, following [7]. We did not try to optimize the algorithm; our goal was not to achieve the best possible accuracy, but to choose a representative algorithm and see the effect on accuracy of various data attribute values. For each user, we form a binary vector $\mathbf{u}$ where $\mathbf{u}[i] = 1$ if the user has visited location i and $\mathbf{u}[i] = 0$ otherwise. From [7], we calculate the cosine similarity between any two users u and v by

$$w_{u,v} = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}.$$

The normalized similarity measure $c_{\mathbf{u},i}$ of u to location i is given by the fraction of users who checked into location i weighted by similarity:

$$c_{\mathbf{u},i} = \frac{\sum_{\mathbf{v} \in L_i} w_{\mathbf{u},\mathbf{v}}}{\sum_{\mathbf{v}} w_{\mathbf{u},\mathbf{v}}},$$

where L_i are the users who have visited location i.

We took the user's top N predicted locations, as given by our similarity measure, and calculated the number of hits, i.e. the number of locations in the test set in this set of N locations. We measured performance using precision and recall. Precision is the fraction of recommendations that are hits; precision@5 means the precision with 5 recommendations. The recall is the number of hits out of the number of possible hits (i.e., the size of the test set for the user). We use the macro-averaged recall, the average over all users of each user's recall. As explained in [7], low precision and recall numbers are expected with such a Top-N recommender. For Austin, our recommendations are an order of magnitude better than random recommendations: five random recommendations would have a precision of approximately 0.005.

User Hardship. We considered two measures of user hardship. In the first, we used KMeans (with two centroids, modeling home and work/school) to cluster each user's training points. We then ranked each user's training points according to their minimum distance to the centroids. In the second, for every point, we calculated the minimum distance to any of the user's other training points. We call the two user hardship measures "KMeans" and "Density". For each measure, we used differential data analysis to rank a user's training points according to the measure: we divided into deciles, and omitted each decile in turn to see the effect on accuracy.

Figure 1 displays our results. We take precision@5 as our proxy for accuracy. We found that recall and precision behaved very similarly in our experiments. Also similar were

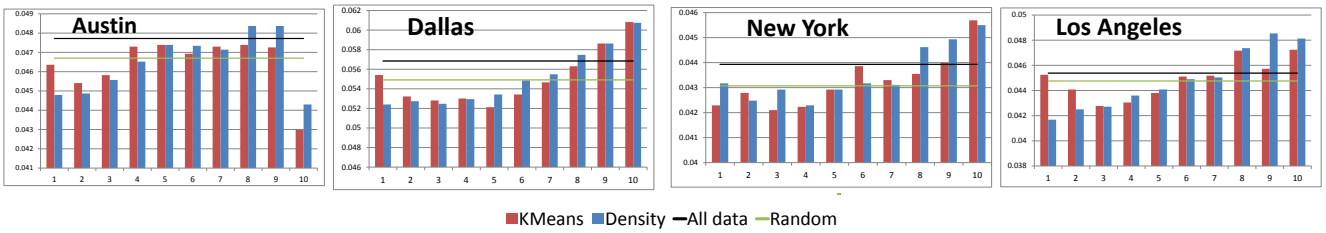


Figure 1: Recommendation accuracy when each decile (measured by user hardship) is omitted from the training set. Two measures of user hardship are shown, using distance to a user's KMeans centroids and using density at the point. Also shown are lines corresponding to accuracy with the entire training set and with random removal of 10% of the data. The figures show that the most useful data for recommendation are generally the first few deciles, i.e. the locations not far from where a user usually spends time.

precision and recall with $N = 10$ and $N = 20$ recommendations.

Observe that user hardship segregates the data well with respect to effect on recommendation accuracy. With the Austin data, we did 20 trials of randomly removing 10% of the training data and computed a mean and standard deviation. The ten sample values of our precision@5 statistic ranged from -3.76 to 2.60 standard deviations away from the mean of the random removals, and only 3 of the 10 sample values were within one standard deviation. In fact, by removing some deciles the accuracy actually became significantly *better* than with all the data, implying these deciles have the effect of noise.

The general trend in the four cities we tried was that the lower hardship deciles (i.e., deciles 2, 3, and 4) were most important for accuracy and higher hardship deciles (i.e., deciles 8, 9, 10) were least important and even could be considered noise. One intuitive explanation is that low- to mid-user hardship is the optimum zone for discovering user preferences. A user would not usually endure high user hardship without other reasons besides just his preferences. Austin is a notable exception for the last decile. The Density user hardship measure generally spreads data points better than KMeans, perhaps owing to the fact that location traces are not defined by only two centroids for many users. We expect the two measures to become closer as the number of KMeans centroids increases.

Timestamp. The Gowalla dataset consists of timestamped checkins, so we also tested whether the timestamp attribute could be used to predict the importance of data for the recommender. These timestamps are in local time of the user. We used our technique of differential data analysis and divided up the day into time intervals so that removal of checkins for each interval corresponds to removing about 10-15% of the data from the training set. We chose this granularity of time interval so one can easily compare with removing 10% of the data randomly or with removing a decile of data using some other attribute. Note that in our experiments we used a binary measure of the user's preference, so removal of a checkin from one time interval does not affect the training set in the case that the user has checked into the same location in a different time interval.

Our results are shown in Figure 2. Overall, timestamp seems less predictive of accuracy than user hardship. The most useful data was around 8 p.m. – midnight, and post-2 a.m. data was least useful. In three out of the four cities, the data from 2 a.m. – 4 a.m. seemed to even confuse the recommender and decreased accuracy. There are also notable differences in the cities, perhaps related to culture and geography (at least for Gowalla users). For instance, the more important and less important data for Los Angeles came a few hours later than for the other cities.

4.2 Stability

In this section we examine the stability of our findings, i.e. whether the relative ranking of deciles will change with new data. We used the Austin data (the city with the most data) and the Density user hardship measure. We performed two experiments, one indicating stability with respect to different test and training sets and one indicating stability with respect to different sets of users. We found that using different sets of users was slightly less stable than using different data sets (from the same set of users), but in either case there was consistency in the less important and more important deciles. We did not study stability over time, although that would be another interesting dimension to study.

In our first experiment, we divided the users into four disjoint sets and with any of the user sets, we get a similar qualitative ranking of the deciles: the last and the closer deciles are important, and the middle deciles through Decile 9 are less important (see Figure 3a). In our second experiment, we examined stability with different sets of data. The resulting plots of accuracy versus decile removed all have similar shapes, but appear to be shifted relative to one another due to the differing test sets (see Figure 3b). We note that for all trials, the four most important deciles were always deciles 1 through 3 and 10. The five least important deciles were always contained in deciles 4 through 9.

4.3 Movie Ratings Dataset

We consider movie recommendation as another case study. We used the well-studied MovieLens 1M dataset [6] which contains 1,000,209 anonymous ratings of 3,952 movies made by 6,040 users. Ratings in MovieLens range from one star to five stars. We divided the dataset into a training set and test set by randomly putting 20% of the ratings for each user in a held-out test set. The remainder is the training set.

We used the Biased Stochastic Gradient Descent (SGD) and Alternating Least Squares (ALS) collaborative filtering algorithms to predict user ratings for movies in our dataset, and Root Mean Squared Error (RMSE) and Mean Average Error (MAE) metrics to measure accuracy. For clarity and space, we present only results with the RMSE metric and the Biased SGD algorithm. Results with MAE and ALS were very similar. Biased SGD and ALS are based on a latent factor model found through matrix factorization. We used 20 factors, so users and movies are each summarized by factor

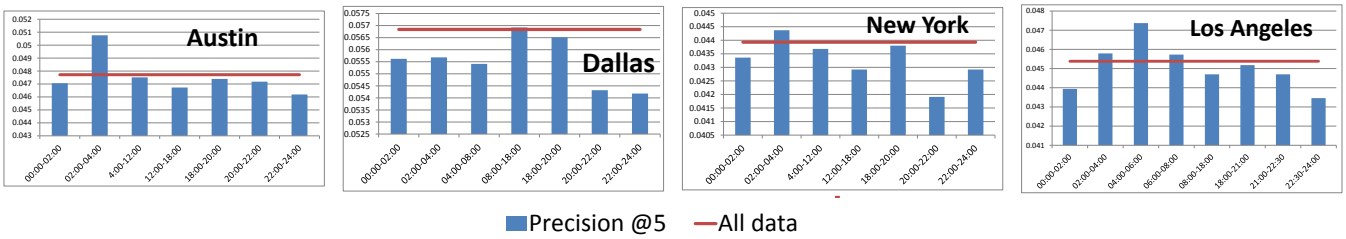
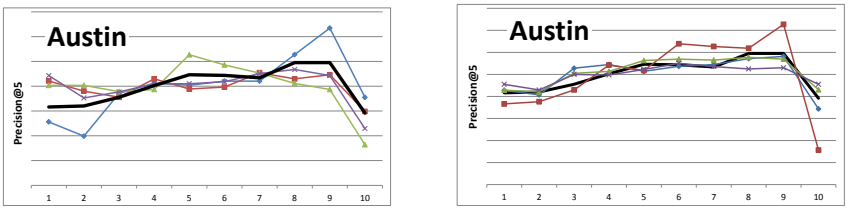


Figure 2: Recommendation accuracy when distinct time intervals are omitted from the Gowalla training set. Each interval corresponds to about 10-15% of all data.



(a) Different user sets (b) Different test and training sets

Figure 3: Stability of Results. For (a), we randomly divided the Austin data into four sets of users, each containing around 1200 users, and computed accuracy per decile removed for each set. Also shown is the same plot for all users, in bold. Since we were only interested in relative values, we centered by vertically translating each plot. For (b), we randomly divided each Austin user's data into 5 equal pieces and repeated our differential data experiment with the user hardship attribute 5 times, with the test set consisting of one of the pieces from each user and the training set consisting of the remaining four pieces. The original training and test set is in bold.

vectors of length 20. We did not search for optimal algorithms or do extensive parameter selection. Again, our goal was not to reduce error but to see which parts of the data had the most effect on error. We used the implementations from [4].

Berkovsky et al. [3] showed previously that high/low ratings were most important to the recommender. We validated their results using our technique of differential data analysis. We divided each user's ratings into 10 equal deciles according to value, so that Decile 1 contains the user's lowest ratings and Decile 10 contains the users high ratings. Figure 4 shows our results when we remove each decile in turn. We ordered each user's ratings in the training set and examined the effect of removing successive rating deciles. For example, when we removed the 10% lowest ratings from each user, we obtained an RMSE of about 0.89. The figure indicates that high and low ratings are most important for recommendation accuracy and that removing deciles 3 or 4 affect accuracy the least.

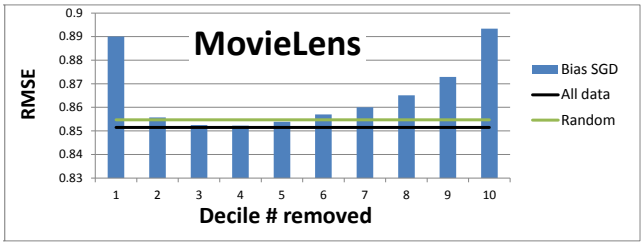


Figure 4: MovieLens Data. Decile 1 contains the user's lowest ratings and Decile 10 contains the user's highest ratings. If necessary, ratings were split randomly across deciles. For example, for a particular user, both decile 1 and decile 2 may contain 1-star ratings; whether a 1-star rating goes in decile 1 or 2 is random. We also show the results with all data (no deciles deleted) and random removal of 10% of the data.

5. CONCLUSION AND FUTURE WORK

We presented a simple technique called differential data analysis. Using a Gowalla checkin dataset and a novel attribute called user hardship, we found that locations closer to a user's usual haunts were most important for recommendation accuracy. It would be interesting to apply the concept of user hardship to other types of data besides location data. For instance, activity data can also be classified according to difficulty or resources required. Using the timestamp attribute, we found that in general very late-night data is least useful and may even confuse the recommender. Interestingly, our results differ from city to city. The root causes for these differences deserve further study. We also applied our techniques to the MovieLens dataset and confirmed previous work that high and low user ratings are most important to the recommender. It would be interesting to explore other data attributes using our technique, for example, to determine which part of the graph for social network data is actually important to the recommender. Finally, our work has applications in user privacy and data reduction; [1] has experiments in this direction, but further study is needed.

6. REFERENCES

[1] Extended version of this paper available on `arXiv.org`.
[2] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *SIGMOD Conference*, pages 439–450, 2000.
[3] S. Berkovsky, Y. Eytani, T. Kuflik, and F. Ricci. Enhancing privacy and preserving accuracy of a distributed collaborative filtering. RecSys '07, pages 9–16, New York, NY, USA, 2007.
[4] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. M. Hellerstein. Graphlab: A new framework for parallel machine learning. *CoRR*, abs/1006.4990, 2010.
[5] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating "word of mouth". In *CHI*, pages 210–217, 1995.
[6] University of Minnesota. Movielens. `http://movielens.umn.edu/`.
[7] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. SIGIR '11, pages 325–334, New York, NY, USA, 2011. ACM.