Predicting Who Rated What in Large-scale Datasets

Yan Liu
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
liuya@us.ibm.com

Zhenzhen Kou
Department of Machine Learning
Carnegie Mellon University
Pittsburgh, PA 15213
zkou@andrew.cmu.edu

ABSTRACT

KDD Cup 2007 focuses on movie rating behaviors. The goal of the task "Who Rated What" is to predict whether "existing" users will review "existing" movies in the future. We cast the task as a link prediction problem and address it via a simple classification approach. Compared with other applications for link prediction, there are two major challenges in our task: (1) the huge size of the Netflix data; (2) the prediction target is complicated by many factors, such as a general decrease of interest in old movies and more tendency to review more movies by Netflix users due to the success of the internet DVD rental industries. We address the first challenge by "selective" subsampling and the second by combining information from the review scores, movie contents and graph topology effectively [1].

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

KDD Cup, Netflix, Link prediction

1. INTRODUCTION

One of the two tasks in KDD Cup 2007 is to predict which users rated which movies in 2006, given the Netflix Prize training data set that contains more than 100 million ratings from over 480 thousand users on nearly 18 thousand movie titles collected between 1998 and 2005. In our practice, we cast the task as a link prediction problem and address it via a simple classification approach.

Link prediction, i.e., the task of predicting the future structure of a network given the current structure, is a fundamental task in many data mining applications, such as social network analysis, protein/genetic interaction prediction, and collaborative filtering recommendation. Many models have been studied and applied to linkage prediction. Network evolution and graph generation models aim to capture how networks grow and change over time, typically based on the topological features [5; 6; 7]. Network evolution and graph generation models focus on abstract graph where no vertex and link attributes are considered. Various relational learning methods have been proposed to define a joint probability over the entire graph - both the node attributes and link structure [8]. Link prediction based on relational learning explores both the link structure and the descriptive attributes of nodes. However, for a huge graph with rich features, the computational cost becomes a problem. Other methods based on binary classification typically require rich features on both nodes and graph structures [3]. One difficulty in applying machine learning algorithms to real problems is the computation expense and feasibility in large-scale applications.

In our approach, we formulate the link prediction as a binary classification problem and solve it via a supervised learning task. To predict a link, we partition the Netflix training set into two non-overlapping subsets - a training set containing ratings appearing before October 2005 and a development set containing ratings after October 2005. A pair of user and movie represents a positive example if there is a rating connection between them, negative otherwise.

There are two major challenges in the KDD Cup "Who Rated What" task: (1) the huge size of the Netflix data; (2) the prediction target is complicated by many factors, such as a general decrease of interest in old movies and more tendency to review more movies by Netflix users due to the success of the internet DVD rental industries. We address the first challenge by "selective" subsampling and the second by combining features based on movie content, review scores, and graph topology effectively and projecting the features over time. We demonstrated that an effective subsampling based on the task requirement is able to provide an effective and efficient solution for a large-scale task. A set of meaningful features has been developed by exploring both the graph topology and movie contents collected from external resources.

2. FEATURE EXTRACTION

To solve a supervised learning problem, effective feature extraction is a must. For our task, we consider two types of features: one is proximity features that represent the similarity in content between the query movie and the movies that the users rated before; the other is the features based on graph topology.

2.1 Content Proximity Features

The movie contents are used to calculate the proximity between a user and a movie. The basic idea is: if a user has rated many animation movies and no horror movies in the

[1]The work is done while Zhenzhen Kou is a summer intern at IBM T.J. Watson Research Center

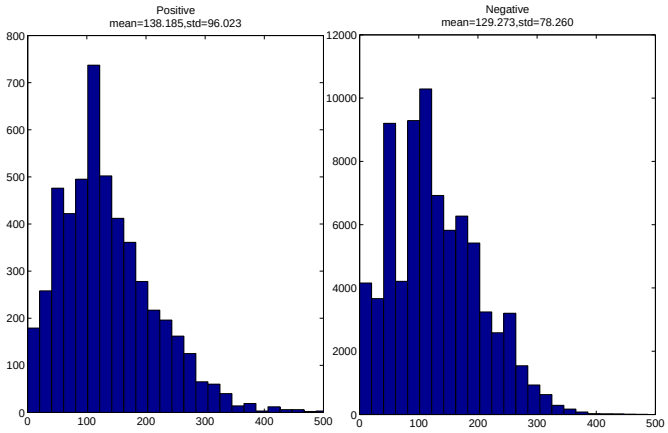

Figure 1: The histograms of proximity feature values from positive examples (left) and negative examples (right)

year of 2005, it is more likely he or she will rate an animation movie rather than a horror movie in 2006. We collect the movie content information, such as plot, director, actor, genre, movie connections from multiple sources on the internet. Then for each movie, we have its content information; for each user, we model the user's preference with contents of movies that have been rated before.

The raw content information that we collect from the internet is unstructured, resulting in a feature set of very high dimensions (over 50,000) when we convert them into structured feature vectors using the bag-of-words representation. It raises a great challenge for any sophisticated classification models to be applied on a data set of extremely high dimensions in both the feature space and example space. Therefore we applied latent semantic indexing (LSI) [1] to obtain a low-dimensional feature representation: for each movie, we constructed one feature set based on directors, actors, genres and so on, and another feature set based on the plots of the movie. Then singular value decomposition (SVD) [2] is applied on the movie-content and movie-plot matrix respectively, and only the top 900 singular vectors are kept for later uses. In this way, each movie is represented with a relatively low-dimensional feature vector so that computing the similarity scores based on dot-product of movie vectors can be executed efficiently.

For each example, i.e., a pair of movie and user, we compute the proximity features in the following way: we use the dot product of two content feature vector to represent the *similarity score* of two movies. For each user, we retrieve the list of movies having been rated and call them *user-related movies*. Given an example, i.e., a movie and a user, we compute all the similarity scores between this movie and the user-related movies, and then use the mean, minimum, and maximum scores as proximity features. To capture the user's preference over time, we project the proximity features into three time ranges - the year of 2003, 2004, and 2005. Figure 1 shows an example of how the proximity features (a larger value mean indicates greater similarity) discriminate between positive and negative examples (different mean, variance as well as the shape of the plots).

2.2 Graph-based Features

Another useful information source is the review history of all users. A graph with users and movies as nodes can be constructed. Consider a graph $G = \langle V, E \rangle$ where each edge $e = \langle u, m \rangle \in E$ represents an interaction between node u and m at a particular time t, i.e., user u rated movie m at time t. There is rich information contained in the graph. One of the most natural measurement for a node is how many other nodes it connects to. In our application, the connectivity represents how popular a movie is, or how active a user is, which is no doubt a meaningful factor in this problem. Therefore the features based on the number of connected edges are used. We also project such features into three time ranges - the year of 2003, 2004, and 2005, to model the trend over time.

Graph topology contains the most important set of features and can be applied to study on all networks. Recent studies on topological features have shown that shortest distance, clustering coefficient, number of common neighbors, and so on are extremely helpful for link prediction. Due to limited time to work on the project, we implemented a set of naive features based on adjacency matrix. In the adjacency matrix, each row is a movie and each column represents a user. Therefore each movie can be represented with the corresponding row in the adjacency matrix. Similarly to our proximity feature, SVD is first applied to convert the vector into a low-dimensional space, the similarity scores between a movie and the user-related movies are then computed, and finally the mean, minimum, and maximum scores are used as features. Figure 2 shows an example of how the graph topology features discriminate between positive and negative examples.

3. EFFECTIVE SAMPLING APPROACH

The first step in data preparation is to collect all the data available. Therefore we combine the Netflix Prize training data and qualification set together as the "Netflix KDD" set. This results in 17,770 movies and 480,189 users, with 103,297,638 reviews. As described on the KDD Cup website, the test sets are generated as follows: the 17770 movies in the Netflix Prize training set were split randomly into two sets, one per task, resulting in 6822 movies for "Who Rated What" task and 8863 movies for "How Many Ratings" task. Let $\mathrm{P}(M^{2006} = i) = p_i$ be the marginal probability that the i^{th} movie is reviewed in 2006, and $\mathrm{P}(U^{2006} = j) = q_j$ be the marginal probability that the j^{th} user reviews a movie in 2006. The movie-user pair (x, y) in the testing set for "Who Rated What" task is generated as follows:

$$x \sim P(M^{2006}), y \sim P(U^{2006}).$$

If user y has reviewed movie x before 2006, the pair (x, y) is removed.

Given the enormous data in the Netflix KDD set, sampling an effective training set is essential to apply any machine learning algorithms. On the other hand, since the goal is to predict the behaviors of existing users in existing movies, it is reasonable to explore similar behaviors in the previous years. Therefore following the sampling methodology as the test data, we generate two sample sets with around 100,000 movie-user pairs for the year 2004 and 2005 as the training data, and two sample sets for the last quarter of year 2005 as the development set. Notice that since we are only interested in the behaviors of existing users to existing movies, the pairs either with the users who joined after 2005 (or 2004)

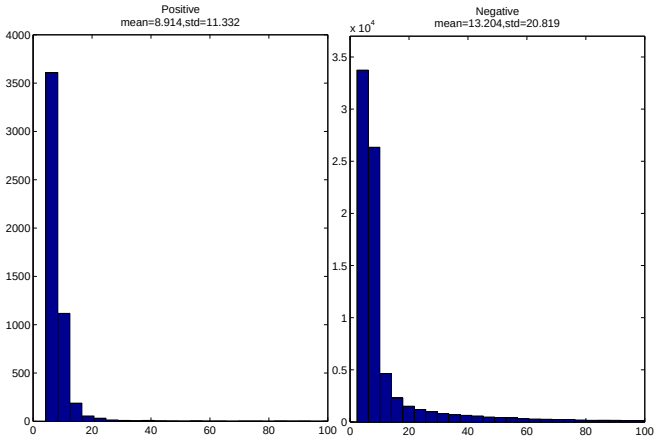

Figure 2: The histograms of graph-based feature values from positive examples (left) and negative examples (right)

or with the movies which are released afterwards, need to be removed. We represent the resulting set as S-2005-1, S-2005-2, S-2004-1, S-2004-2, S-2005Q4-1, S-2005Q4-2 respectively. The positive rate of the true labels in the sampled sets from different years is shown in Table 1. As we can see, the ratio remains similar over the years. In particular, the estimated rate in 2005 is very close the one in the KDD Cup test set, which serves an excellent training set for our later prediction.

Year	2004	2005	Q4 2005	2006
Positive rate	6.83%	7.94%	6.12%	7.80%

Table 1: Positive rate of the true labels in the sampled sets from year 2004, 2005, the last quarter of 2005, and 2006 (from the released answer set posted on the KDD Cup website)

4. LEARNING ALGORITHMS

Even though we have only sampled a subset of the huge training set, the number of examples is still around 100,000, which renders useless many sophisticated classification algorithms, such as support vector machines. After careful examination of the characteristics of the data, including : (1) an imbalanced set with only 6-8% positive examples; (2) heterogenous attributes, i.e. the features are gathered and extracted from multiple information sources, such as movie content, review information and so on; (3) huge number of training and testing examples, we explore the following learning strategies, including:

Single classifier: a straightforward solution to our task as a classification problem, is to apply some classifiers, which have been studied extensively over the past ten years. In our experiment, we have tried logistic regression, support vector machines (which fails to converge during the training phase and generate any reasonable predictions), decision trees, and ridge regression [4]. We use the toolkit Matlabarsenal [2], which is an open-source MATLAB package that encapsulates most of popular classification algorithms to facilitate the research efforts on developing and evaluating classification algorithms on real-world data sets. In our experiments, we find that ridge regression and logistic regression are efficient and provide most accurate results. In our submission, we use the ridge regression since its optimization criterion agrees with the evaluation measure, i.e. root mean squared errors (RMSE).

Ensemble of classifiers: In addition to the simple single classifiers, we also examine two ensemble approaches, including (1) building sub-classifiers on subsets of training examples to alleviate the problem of too many examples; and (2) building separate classifiers for each set of features (from the same sources) to reduce the dimension of the raw features. To combine the prediction of the sub-classifiers, we use the pre-set weights (by human) and the weights that are learned from the other development set. In our experiment, we find that the first ensemble approach, i.e. building classifiers on subsets of training examples, perform consistently worse than building a single classifier on the whole training set. Therefore, we focus on the second approach with different combination strategies.

5. EXPERIMENT RESULTS

In this section, we describe the experiment results from two settings: one is the validation setting, which is used for feature selection and classifier selection; the other is submission setting, which reports our final submitted results for KDD Cup 2007 "Who Rated What" task.

5.0.0.1 *Validation Results.*

In our validation experiments, we use S-2005Q4-1 as the testing set, S-2005Q4-2 as the development set and S-2005-1, S-2005-2, S-2004-1, S-2004-2 as the training sets. The root mean squared error (RMSE) is used for evaluation measure. The results are shown in Figure 3. The RMSE of the baseline method, which is calculated by assigning all the examples as the prior of the test set, is 0.2394. From the results, we can see that: (1) all the methods achieve better results than the baseline method, which requires the nontrivial estimate of the prior of the positive examples in the test set; (2) the training sets from year 2005 are much more effective than those from 2004; (3) the ensemble approach, which learns the weights from the development set using logistic regression, perform consistently the best.

5.0.0.2 *Submission Results.*

For KDD cup 2007, we target at answering the question of "Who Rated What" in 2006. In the validation experiments, we have observed that the training sets sampled from the current year (compared with other years) are the most effective for the predictions of the next year. Therefore we use the sets sampled from year 2005 as the training data. In addition, to reduce the variance, we build two models using the two 2005 sets, i.e. S-2005-1 and S-2005-2 respectively, and then average the predictions are with equal weights. Before the deadline of the submission, we generate the results on the three models we have examined before but only submit the one generated by the single classifier (with ridge regression) using all the features in order to avoid possible overfitting of the ensemble approach (whose weights are learned using S-2005Q4-2 as the development set since we

[2]http://finalfantasyxi.inf.cs.cmu.edu/MATLABArsenal/MATLABArsenal.htm

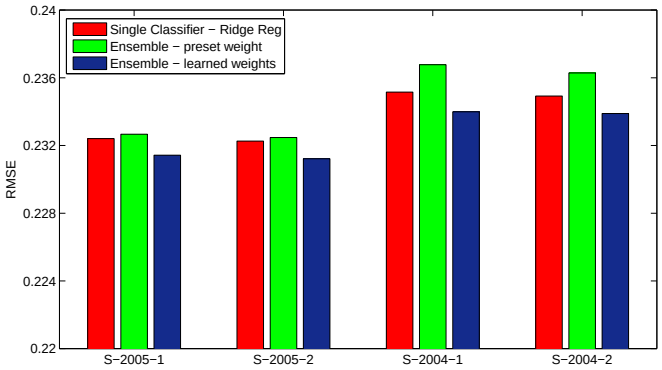

Figure 3: RMSE of the validation experiments using S-2005Q4-1 as the testing set and S-2005Q4-2 as the development set

do not have any data from 2006). Table 2 shows the RMSE of the three methods on the KDD Cup test set after the answer set is released and Table 3 lists the RMSE results of the top performers in this task.

Method	Baseline	Single Classifier	Ensemble: preset wgt	Ensemble: learned wgt
RMSE	0.268	**0.265**	0.266	0.263

Table 2: RMSE of our models on KDD Cup test set. The predictions from single classifier are submitted (in bold letters).

Team	RMSE
Hungarian Academy of Sciences	0.256
Neo Metrics	0.263
IBM Research	0.265
# 4	0.267
Baseline	0.268

Table 3: RMSE of top performers on KDD Cup "Who Rated What" task.

6. DISCUSSION

The KDD Cup 2007 focuses on the Netflix Prize data with rich information of the movie reviews and it has attracted the attention of many researchers and scholars in related areas. There are several observations that we find interesting in both future research directions and industrial applications:

1. *Correct sampling is essential*: The Netflix Prize data is a typical example of the data we need to handle in many real applications. A recent trend in the research of machine learning and data mining is to adapt the successful but complex algorithms for large-scale applications. In the exercise of KDD Cup, we have demonstrated that an effective subsampling based on the task analysis is able to provide an accurate and efficient solution.

2. *The effectiveness of multi-task learning*: The KDD Cup 2007 uses the same data as the Netflix Prize competition, but focuses on different tasks. An initial thought we have for the "Who Rated What" task is to explore the features or lessons that have been examined in the Netflix Prize competition. However, we fail to achieve any significant progress on that direction: the models trained on the SVD features from the Netflix Prize competition cannot even help us beat the baseline. As we can see, a naive sharing of feature space is far from getting the most of multi-task learning. It would be interesting to explore how to effectively make use of sharing information in different tasks for learning.

Acknowledgements

We want to thank Rick Lawrence, Naoki Abe, Prem Melville, Hisashi Kashima, Shohei Hido, Chandan Reddy, Yi Zhang, Hanghang Tong, Rong Yan, Yuan Yuan, and many others for useful discussion.

7. REFERENCES

[1] M. Berry, S. Dumais, and T. Letsche. Computational methods for intelligent information access. In *Proc. of Supercomputing'95*, 1995.

[2] G. Golub and C. V. Loan. *Matrix Computations*. Johns-Hopkins, Baltimore, second ed., 1989.

[3] M. Hasan, V. Chaoji, S. Salem, and M. J. Zaki. Link prediction using supervised learning. In *Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.

[4] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer; 1st ed., 2001.

[5] Z. Huang. Link prediction based on graph topology: The predictive value of generalized clustering coefficient. In *Proc. of KDD 06 workshop on Link Analysis: Dynamics and Static of Large Networks (LinkKDD2006)*, 2006.

[6] H. Kashima and N. Abe. A parameterized probabilistic model of network evolution for supervised link prediction. In *Proc. of the 2006 IEEE International Conference on Data Mining (ICDM 2006)*, 2006.

[7] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proc. of 12th International Conference on Information and Knowledge Management (CIKM)*, 2003.

[8] B. Taskar, M.-F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *Proc. of Advances in Neural Information Processing Systems (NIPS 2003)*, 2003.