International Journal of Management Science
Vol 16, No 2, Special Issue, September 2010

Determining Absolute Interpolation Weights for Neighborhood-Based Collaborative Filtering

Hyoung Do Kim
School of Business Administration, Hanyang Cyber University, Seoul, 133-791, Rep. of Korea

(Received: March 15, 2008 / Revised: December 10, 2009 / Accepted: December 20, 2009)

ABSTRACT

Despite the overall success of neighbor-based CF methods, there are some fundamental questions about neighbor selection and prediction mechanism including arbitrary similarity, over-fitting interpolation weights, no trust consideration between neighbours, etc. This paper proposes a simple method to compute absolute interpolation weights based on similarity values. In order to supplement the method, two schemes are additionally devised for high-quality neighbour selection and trust metrics based on co-ratings. The former requires that one or more neighbour's similarity should be better than a pre-specified level which is higher than the minimum level. The latter gives higher trust to neighbours that have more co-ratings. Experimental results show that the proposed method outperforms the pure IBCF by about 8% improvement. Furthermore, it can be easily combined with other predictors for achieving better prediction quality.

Keywords: Collaborative Filtering, Interpolation Weight, Neighbourhood Selection

1. Introduction

Collaborative filtering (CF) methods based on neighbours are very popular due to their intuitiveness, absence of the need to train and tune many parameters, and the ability to easily explain to a user the reasoning behind a recommendation. The basic idea is to make recommendations based upon ratings that users have assigned to products. Therefore, it is very important to select the neighbors properly to improve the prediction quality. There have been many ways to select proper neighbors using similarity measures such as Pearson's correlation. Predictions are then made by

* Corresponding author, E- mail: hdkim@hycu.ac.kr

weighting the rating values of the selected neighbours.

Despite the overall success of neighbour-based CF methods, there are some fundamental questions about neighbour selection and prediction mechanisms including arbitrary similarity, over-fitting interpolation weights, no trust consideration for neighbours, and so on (Bell and Koren [3]). Given a set of neighbors, interpolation weights are computed by dividing respective similarity value by the summation of all the similarity values. That is, the interpolation weights sum to one. Such a relative weighting scheme can cause many problems. Assuming that a particular item is predicted by a subset of the neighbors, it is impossible for the subset to receive all the weights from the neighbors truly needed. Furthermore, the interpolation weights may result in over-fitting. Suppose that an item has no useful neighbors rated by a particular user. In that case, it would be best to ignore the neighborhood information, staying with the current normalized data. Nevertheless, current CF methods usually employ a weighted average of ratings for the uninformative neighbors.

In order to overcome the limitations, this paper proposes a simple method to determine absolute interpolation weights whose sum is not one. In order to systematically support the method, two schemes are additionally devised for high-quality neighbour selection and trust consideration. The former requires that more than one neighbour's similarity should be better than a pre-specified high level. The latter gives high trust to neighbours that have more co-ratings with the active user or item.

The rest of the paper is organized as follows. Section 2 summarizes related research about neighbour-based CF with introduction to basic notations. Section 3 proposes the simple method and two additional schemes. In Section 4, experimental results are presented. Finally, Section 5 concludes the paper with future research directions.

2. Related Research

Recommender systems analyze patterns of user interest in items (products) to provide personalized recommendations of items that will suit a user's (customer's) taste [1]. Broadly speaking, they use either of two strategies: content-based approach versus collaborative filtering [3]. The former profiles each customer or product, allowing programs to associate customers with matching products. The latter relies on past customers' behaviour and does not require the creation of explicit profiles. CF analyzes relationships between users and interdependencies among items, in order to identify new user-item associations. CF requires no domain knowledge as well as avoiding the need for extensive data collection about items and/or users. Furthermore, CF offers the potential to uncover patterns that would be difficult or impossible to profile using the former approach.

Various approaches developed in the field can be classified into two general categories: memory-based CF and model-based CF [4]. Memory-based collaborative filtering uses a similarity measure between pairs of users to build a prediction, typically through a weighted average. The chosen similarity measure determines the accuracy of the prediction and numerous alternatives have been studied [7]. Some potential drawbacks of memory-based CF include scalability and sensitivity to data sparseness. In general, schemes that rely on similarities across users cannot be precomputed for fast online queries. Another critical issue is that memory-based schemes must compute a similarity measure between users. There are also many model-based approaches to CF [12]. They are based on linear algebra or on AI techniques such as neural networks and clustering. In comparison with memory-based schemes, model-based CF algorithms are typically faster at query time though they might have expensive learning or updating phases. Model-based schemes can be preferable to memory-based schemes when query speed is crucial.

The most common form of memory-based CF is the neighborhood-based approach [15]. The approach identifies pairs of users or items that tend to be rated by similarly-minded users with similar histories of rating in order to predict ratings for unobserved user-item pairs. Methods and schemes of this approach can be classified into two groups: user-based collaborative filtering (UBCF) and item-based collaborative filtering (IBCF).

Neighbourhood-based CF methods generally include functional components such as data normalization, neighbor selection, and determination of interpolation weights. It is customary for the first component to normalize data before activating a neighbourhood-based method. Usually this is achieved by adjusting for the varying mean ratings across users and/or items. The second component relates users or items by various heuristic variants of correlation coefficients, which allow direct interpolation from the neighbors' scores by the third component.

Since the basic idea of neighbourhood-based CF is to make recommendations based upon similar items' or users' ratings, it is very important to select the neighbours properly to improve the prediction quality. Many similarity measures have been proposed for neighbourhood-based CF systems. Cosine measure looks at the angle between two vectors of ratings where a smaller angle is regarded as implying greater similarity. Equation (1) defines the cosine similarity. Here, s_{ij} identifies the similarity of two items *i* and *j*, r_{ui} the value rated by user *u* about item *i*. Pearson's correlation measures the linear correlation between two vectors of ratings as equation (2) defines. Adjusted cosine similarity is used for some ICBF methods, where the difference in each user's rating scale is taken into account as equation (3) shows [16]. Constrained Pearson's correlation is a slightly modified version of Pearson's correlation, which allows only the pairs of ratings on the same side to contribute to the correlation [17]. In equation (4), r_{med} is the median value of the rating scale to determine whether the pairs of the ratings are on the same side.

$$s_{ij} = \frac{\sum_u r_{ui} r_{uj}}{\sqrt{\sum_u r_{ui}^2 \sum_u r_{uj}^2}} \tag{1}$$

$$s_{ij} = \frac{\sum_u (r_{ui} - \overline{r}_i)^2 (r_{uj} - \overline{r}_j)^2}{\sqrt{\sum_u (r_{ui} - \overline{r}_i)^2 \sum_u (r_{uj} - \overline{r}_j)^2}} \tag{2}$$

$$s_{ij} = \frac{\sum_u (r_{ui} - \overline{r}_u)^2 (r_{uj} - \overline{r}_u)^2}{\sqrt{\sum_u (r_{ui} - \overline{r}_u)^2 \sum_u (r_{uj} - \overline{r}_u)^2}} \tag{3}$$

$$s_{ij} = \frac{\sum_u (r_{ui} - r_{med})^2 (r_{uj} - r_{med})^2}{\sqrt{\sum_u (r_{ui} - r_{med})^2 \sum_u (r_{uj} - r_{med})^2}} \tag{4}$$

Note that IBCF methods deliver better quality estimates than UBCF ones, while allowing more efficient computations [16]. The greater efficiency occurs because the number of items is significantly lower than the number of users, which allows precomputing all item-item similarities for retrieval as needed. In this respect, the rest of this paper considers mainly from the viewpoint of IBCF.

Assuming consine similarity, for instance, predicted value p_{ui} for the rating to be made by user u for item i is then computed by equation (5) using the similarity values. Note that interpolation weights are determined by the relative weights of the similarity values, summing to one.

$$p_{ui} = \frac{\sum_j s_{ij} r_{uj}}{\sum_j s_{ij}} \tag{5}$$

In order to remove item- and user-specific biases that may prevent the model from revealing the more fundamental relationship, some baseline predictors, b, can be employed as in equation (6). Usually b_{ui} takes the mean rating of item i for equation (2), the mean rating of user u for equation (3), and the median value r_{med} for equation (4), respectively.

$$p_{ui} = b_{ui} + \frac{\sum_j s_{ij}(r_{uj} - b_{uj})}{\sum_j s_{ij}} \tag{6}$$

Despite the overall success of neighborhood-based CF methods, there have been a few questions asked if the traditional measures are suitable. The questions result in other similarity measures such as Spearman rank correlation, entropy-based uncertainty measure, etc. In Spearman rank correlation, similarity is computed using ranks instead of rating values. The large number of tied rankings results in a degradation of the prediction accuracy [8]. The measure based on entropy uses conditional probability techniques to measure the reduction in entropy of the active user's ratings that results from knowing the another user's ratings [10]. A heuristic similarity measure based on the minute meanings of co-ratings defines similarity as the multiplication of three factors: proximity, impact, and popularity [2].

3. Determining Interpolation Weights

This paper proposes a simple method for improving the determination of interpolation weights. Given a set of neighbors, each interpolation weight is computed as the ratio of the similarity value s_{ij} over the summation of all the similarity values,

$s_{ij} \Big/ \sum_j s_{ij}$. Therefore, the summation of all the interpolation weights for the neighbors of the active item/user is one. Such a relative weighting scheme can cause many problems. Assuming that a particular item is predicted by a subset of the neighbors, it is impossible for the subset to receive all the weights truly needed. Each similarity value between an item and one of its neighbors is computed independently of the other neighbors. Furthermore, the interpolation weights may result in over-fitting. Suppose that an item has no useful neighbors rated by a particular user. In that case, it would be best to ignore the neighborhood information, staying with the current normalized data. Nevertheless, current neighborhood-based CF methods usually employ a weighted average of ratings for the uninformative neighbors.

Let's have an example. Figure 1 shows a simple user-item rating matrix. If one is to estimate r_{32}, the rating of Item2 by User 3, he/she needs to calculate similarity values. Assuming IBCF and cosine similarity, potential neighbors of Item 2 are Item1, Item3 and Item4. Similarity values between Item2 and the potential neighbors are $s_{21} = 0.16$, $s_{23} = 0.23$, and $s_{24} = 0.13$, respectively. If nearest-2 neighbor selection scheme is applied, item1 and item3 are its neighbors selected. The interpolation weights are then determined as the relative weights of s_{21} and s_{23} compared with the sum of all the similarity values. That is, interpolation weights for Item1 and Item3 are $s_{21}/(s_{21}+s_{23})$ = 0.41 and $s_{23}/(s_{21}+s_{23})$ = 0.59. In spite of the low similarity of Item1 and Item3 to Item2, high interpolation weights are assigned to them. If we apply threshold-based neighbor selection scheme to this example, e.g. minimum similarity threshold $\theta = 0.2$, then Item3 is selected as the only neighbor. As a result, the interpolation weight of Item3 to Item2 is 1.0.

User\Item	Item1	Item2	Item3	Item4	Sum	Average
User1	3	5		2	10	3.3
User2		4	2	4	10	3.3
User3	4	?	1	3	8	2.7
User4	5	3	3	5	16	4
Sum	12	12	6	14	44	
Average	4	4	2	3.5		3.4

Figure 1. Neighborhood Selection Example

One solution to the problem is to get the interpolation weights directly, which are then used as the following equation:

$$p_{ui} = \sum_j w_{ij} r_{uj} \tag{7}$$

One method to determine the interpolation weights is through formulating a least squares problem:

$$\min_w \sum_v (r_{vi} - p_{vi})^2 = \min_w \sum_v (r_{vi} - \sum_j w_{ij} r_{vj})^2 \tag{8}$$

We may solve the problem as a linear equation system of the form Aw = b, which incurs high cost of computations and retains tight assumptions for preventing A to be singular [3]. Instead of the complex linear equation system, this paper defines w_{ij} as follows:

$$w_{ij} = \alpha_i s_{ij} \Big/ \sum_j s_{ij} \tag{9}$$

That is, the problem is simplified by accepting the relative weights based on similarity values. The least squares problem is then converted into the following:

$$\min_{\alpha_i} \sum_v (r_{vi} - \sum_j \alpha_i s_{ij} r_{vj} \Big/ \sum_j s_{ij})^2 \tag{10}$$

α_i is the only variable to be determined in equation (10). We can get an optimal estimate of α_i by the following equation:

$$\alpha_i = \frac{\sum_v r_{vi}}{\sum_v \sum_j \frac{s_{ij} r_{vj}}{\sum_k s_{ik}}} = \frac{\sum_v r_{vi}}{\sum_j (\sum_v r_{vj})(\frac{s_{ij}}{\sum_k s_{ik}})} \tag{11}$$

In order to prevent over-fitting, basic predictors such as item averages can be applied to equation (11) as the estimate for each item j, which has not been rated by a user v who has rated item i. Let's get back to the example of Figure 1. The interpolation weights of Item 1 and Item3 can be calculated as follows:

i) The values for r_{21} and r_{13} are predicted using item averages. $p_{21} = 4$ and $p_{13} = 2$, when we are using item averages as the predictors.
ii) α_i is then calculated by the equation (11). Here, $\alpha_2 = (5+4+4)/\{(3+4+5)\text{x}0.41+(2+2+3)\text{x}0.59\} = 13/9.05 = 1.44$.
iii) The absolute interpolation weight w_{ij} is then determined by the equation (9). $w_{21} = 1.44\text{x}0.41 = 0.59$ and $w_{23} = 1.44\text{x}0.59 = 0.85$.
iv) The prediction value p_{32} is finally estimated by 4x0.59+1x0.85 = 3.21. Please compare the estimate with the original one by IBCF, 4x0.41+1x0.59 = 2.23.

The weighting mechanism determines how much can be contributed from the neighbors' rating values. However, the interpolation weights cannot be guaranteed to be small when an item has only weak neighbors. In order to remove this situation, another requirement is added for neighbor selection: one or more neighbors should have similarity values better than a specified level called 'minMax,' regardless of the nearest-N and minimum threshold schemes.

In applying the method, one important issue is the amount of trust to be placed in a correlation with a neighbor. It is common for an active user or item to have highly correlated neighbors that were based on a very small number of co-ratings. These frequently proved themselves to be terrible predictors for the active user or item. For example, Herlocker *et al*. [7] applied a significance weight of n/50, where n is the number of co-ratings. If there were more than 50 co-ratings, then a significance weight of 1 was applied. In order to improve this aspect, we define trust as the form of $n_{ij}/(n_{ij}+\beta)$, where n_{ij} is the number of co-ratings of users i and j. Similarity values are then redefined as follows:

$$s_{ij} = \frac{\sum_u r_{ui} r_{uj}}{\sqrt{\sum_u r_{ui}^2 \sum_v r_{uj}^2}} \bullet \frac{n_{ij}}{(n_{ij} + \beta)} \tag{12}$$

4. EXperimentAL Results

The proposed method has been simulated based on the Netflix dataset [13], a comprehensive dataset including more than 100 million movie ratings published by Netflix Inc. A large challenge for prediction is posed by the three characteristics of the dataset: size, sparseness and non-randomness [3]. The ratings in scale of 1 to 5 were performed by 480189 users on 17770 movies. Each rating is associated with a date between mid 1999 and the end of 2005. The size requires more efficient methods in computing. Even with over 100 million ratings, almost 99% of the potential user-item pairs have no ratings. Consequently, it is hard to apply machine-learning methods designed for complete data situations to the data. Furthermore, the statistical pattern of observed data is very non-random. This fact complicates the challenge to detect weak signals for users/movies with sufficient samples while avoiding over-fitting with very few ratings.

For performance evaluation of the mechanism, root mean squared error (RMSE) and mean absolute error (MAE) are used as the statistical accuracy metrics. RMSE and MAE are computed by (13) and (14), respectively. In the equations, M is the number of users and N is the number of items.

$$RMSE = \sqrt{\frac{\sum_{u=1}^{M}\sum_{j=1}^{N}(p_{ui} - r_{ui})^2}{MN}} \tag{13}$$

$$MAE = \frac{\sum_{u=1}^{M}\sum_{j=1}^{N}\left|p_{ui} - r_{ui}\right|}{MN} \tag{14}$$

The experimental results are shown in Table 1. Randomly selected 1% movie ratings for randomly selected 0.01% users are used as test data in each experiment, performed ten times for each parameter setting. IBCF with absolute weighting (AW) outperforms the pure IBCF by about 8% improvement in MAE with parameter setting: Neighbour Size NS = 100, Minimum Threshold θ = 0.1 (the statistic value of its Mann-Whitney test for RMSE is 18.0 (p-value 0.016). However, IBCF with minMax and IBCF with Trust don't show fundamental improvement compared with the pure IBCF. The two schemes cannot affect the selection process because there are enough similar items that have high-level similarity in the Netflix data. IBCF with AW and Trust and IBCF with AW, Trust and minMax show only slight improvement over IBCF with AW due to the same reason.

Table 1. Experimental Results of IBCF with AW

Methods / Schemes	Parameters	RMSE	MAE
Pure IBCF	NS = 100, θ = 0.1	1.093	0.857
	NS = 100, θ = 0.9	1.100	0.859
IBCF with AW	NS = 100, θ = 0.1	0.982	0.780
	NS = 100, θ = 0.5	0.982	0.780
	NS = 100, θ = 0.9	0.986	0.784
IBCF with minMax	NS = 100, θ = 0.1, minMax = 0.5	1.094	0.857
	NS = 100, θ = 0.7, minMax = 0.9	1.094	0.857
IBCF with Trust	NS = 100, θ = 0.1, β = 50	1.068	0.842
	NS = 100, θ = 0.1, β = 100	1.064	0.841
	NS = 100, θ = 0.7, β = 200	1.059	0.836
IBCF with AW, Trust	NS = 100, θ = 0.1, β = 100	0.972	0.777
	NS = 100, θ = 0.5, β = 100	0.981	0.777
	NS = 100, θ = 0.7, β = 100	1.003	0.803
	NS = 100, θ = 0.9, β = 100	1.057	0.945
IBCF with AW, minMax, Trust	NS = 100, θ = 0.1, minMax = 0.5, β = 100	0.969	0.773
	NS = 100, θ = 0.3, minMax = 0.7, β = 100	0.994	0.795
	NS = 100, θ = 0.5, minMax = 0.9, β = 100	1.028	0.829

As a model-based approach, latent factor models seek for the more holistic goal to uncover latent features that explain the observed ratings. For example, singular vector decomposition (SVD) associates each user u with a user-factors vector p_u and each movie with a movie-factors vector q_i. Predictions are inner products of the form $r_{ui} = p_u^T q_i$. Theoretically, these vectors capture the f-most prominent features of the data, leaving out less significant patterns in the observed data that might be mere noise. For sparse data, alternative estimation methods are required in order to deal with the missing elements and to avoid over-fitting. Regularized SVD is to find matrices p_u and q_i by solving the following optimization problem, where $\lambda > 0$ is a regularization parameter [6, 11, 14, 18].

$$\min\left\{\frac{\lambda}{2}\left(\sum_{uk} p_{uk}^2 + \sum_{ik} q_{ik}^2\right) + \sum_{ui}\left(r_{ui} - \sum_{k} p_{uk} q_{ik}\right)^2\right\} \tag{15}$$

Using regularized SVD for CF is to fit a latent factor model to the observed ratings r_{ui}, such that for the known rating values r_{ui}, the corresponding elements p_u and q_i are close to r_{ui}. The regularization term is to restrict the domains of p_u and q_i in order to prevent over fitting, so that the resulting model has good generalization performance.

This paper employs the estimates resulted from a regularized SVD model as b_{ui} in equation (6). The residual term (r_{uj}-b_{uj}) is then used for calculating similarity values and absolute interpolation weights. The experimental results are summarized in Table 2, where small improvement is achieved over the regularized SVD.

Table 2. Experimental Results of Combining SVD and IBCF with AW

Methods / Schemes	Parameters	RMSE	MAE
SVD	λ = 0.015	0.926	0.732
SVD and IBCF with AW	λ = 0.015, NS = 100, θ = 0.1	0.918	0.724

5. Conclusion

This paper has presented an absolute weighting method for neighbourhood-based CF. The sum of the interpolation weights is not restricted to one. The interpolation weights of similar items are determined using the rating values of users that have rated the active item, from the viewpoint of ICBF. Although determination of absolute interpolation weights is based on similarity values, the IBCF with absolute weighting outperforms the pure IBCF by about 8% improvement in MAE. However, IBCF with minMax and IBCF with Trust just show a little improvement compared with the pure IBCF. The reason is that they have few chances to affect the selection process due to the abundant similar items in the dataset. The method is also applied to the result of a regularized SVD for removing global effects [5]. The combination demonstrates its potential.

Future research directs toward two targets: One is how to explicitly account for complex relationships among neighbours. We need to account for the interactions among neighbors since each similarity value between an item and one of its neighbors is computed independently of the other neighbors. The other is how to combine the weighting scheme with other methods for better improvement.

References

[1] Adomavicius, G. and A. Tuzhilin, "Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (2005), 734-749.

[2] Ahn, H. J., "A New Similarity Measure for Collaborative Filtering to Alleviate the New User Cold-Starting Problem," *Information Sciences* 178, 1 (2008), 37-51.

[3] Bell, R. M. and Y. Koren, "Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights," *Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM)* (2007), 43-52.

[4] Breese, J. S., D. Heckerman, and C. M. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, Madison, Wisconsin, USA, (1998), 43-52.

[5] Burke, R., "Hybrid Recommender Systems: Survery and Experiments," *User Modeling and User-Adapted Interaction* 12, 4 (2002), 331-370.

[6] Funk, S., "Netflix Update: Try This at Home," http://sifter.org/~simon/juornal/2001211.html.

[7] Herlocker, J. L., J. A. Konstan, L. G. Terveen, and J. Riedl, "An Algorithmic Framework for Performing Collaborative Filtering," *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, USA, (1999), 230-237.

[8] Herlocker, J. L., J. A. Konstan, A. Borchers, and J. Riedl, "Evaluating Collaborative Filtering Recommender Systems," *ACM Transactions on Information Systems* 22 (2004), 5-53.

[9] Hofmann, T., "Latent Semantic Models for Collaborative Filtering," *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 89-115.

[10] Kim, H. D., "Collaborative Filtering by Consistency-Based Trust Definition," *Proceedings of the Joint Autumn Conference of the Korea Society of Information Technology Applications and Other Societies*, Chungju, Rep. of Korea, (2007), 551-556.

[11] Kurucz, M., A. A. Benczur, and K. Csalogany, "Methods for Large Scale SVD with Missing Values," *Proceedings of KDD-Cup and Workshop at the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Jose, CA, USA, (2007), 31-38.

[12] Lemire, D. and A. Maclachlan, "Slope One Predictors for Online Rating-Based Collaborative Filtering," *Proceedings of the SIAM Data Mining (SDM'05)*, Newport Beach, USA, (2005), 471-476.

[13] Netflix, "Netflix Movie Dataset," http://www.netflixprize.com/.

[14] Paterek, A., "Improving Regularized Singular Value Decomposition for Collaborative Filtering," *Proceedings of KDD-Cup and Workshop at the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Jose, CA, USA, (2007), 39-42.

[15] Sandvig, J. J., B. Mobasher, and R. Burke, "A Survey of Collaborative Recommendation and the Robustness of Model-Based Algorithms," *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* 31, 2 (2008), 3-13.

[16] Sarwar, B. M., G. Karypis, J. A. Konstan, and J. Riedl, "Item-Based Collaborative Filtering Recommendation Algorithms," *Proceedings of the 10th Int'l Conference on the World Wide Web*, Hong Kong, (2001), 285-295.

[17] Shardanand, U. and P. Maes, "Social Information Filtering: Algorithms for Automating Word of Mouth," *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, Denver, CO, USA, (1995), 210-217.

[18] Wu, M., "Collaborative Filtering via Ensembles of Matrix Factorizations," *Proceedings of KDD-Cup and Workshop at the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Jose, CA, USA, (2007), 39-42.