Category Classification Using Relational Data

Ziqi Pang, Naiqing Guan, Xiuping Cui, Pengfei Wang
Peking University
pangziqi, guannq, 1600013028, 1600012889@pku.edu.cn

ABSTRACT

Relational data consists of both the local attributes and mutual relations among the entities. Comparing to data with only local attributes, it offers the potential to improve the accuracy on many applications like classification.

In this paper, we introduce a method dealing with the relational data using Conditional Random Field and ensemble methods. To accomplish this, we present a model that leverages the relations of data and then combines different classifiers with ensemble. On a public data set, our method shows competitive performance ti state-of-art approach. Moreover, we discuss some further issues on model design and implementation, including asymmetry of factors in Markov Random Field, methods to improve models further, etc.

Keywords

Network data, Markov Random Field, Ensemble

1. INTRODUCTION

Relational data is ubiquitous and they are arranged in the form of networks to reflecting the relations among the entities. This brings about both potential and difficulties in applications. The potential comes from the additional information in the connections among the nodes, and difficulties come from dealing with the complexity of the network structure. At the same time, utilizing the relations in the network becomes crucial for tasks like information retrieval and classification as many approaches only dealing with the local attributes of entities have already reached their peaks. One of the representative problem in this area is classifying the categories of academic papers with their citation and co-author relations. The paper-category-classification problem consists of the main components in most similar problems, *i.e.* local attributes, connections among entities, different kinds of relations. In this paper, we will demonstrate an algorithm focusing on this paper category classification problem and evaluate its performance on the CORA[14] data set.

Currently, there are mainly two kinds of approaches to deal with the relational data. One of them focuses on the local structure of the network, called collective classification[11], and the representative algorithms are Iterative Classification Algorithm(ICA)[16] and Gibbs Sampling(GS)[9]. The other kind of approaches tend to model the network structure in a global way, including Markov Random Field[25], Conditional Random Field[8] and other similar graphical probability models. We use conditional random field in our paper.

Another important issue in the classification problem is the existence of different kinds of relations. Therefore, combining the classifiers originating from different features and relations is necessary. Accordingly, a group of algorithms called "ensemble" are widely used. Ensemble is a classical problem in machine learning[4]. In our paper, we will discuss two ensemble methods and compare their performance.

We make the following contributions in this paper:

1. We apply conditional random field on a network data set(dealing with citation and co-author relationship) with taking missing labels and the asymmetry of factors into consideration.

2. To combine different relations, we try voting and neural networks for ensemble, compare their performance and analyze the cause. To the best of our knowledge, we are the first to use neural network for ensemble in this kind of problem.

3. The algorithm in our paper shows competitive performance compared to many previous algorithms.

4. In order to evaluate the performance of algorithms under different situations, we try different proportion of the training set and compare the accuracy of the classification algorithm.

2. RELATED WORK

One of the earliest approaches considering relations between entities was [1], which contains a probabilistic model for classification of web pages using the content of the pages and the labels of linked pages. It is proved that, the classification accuracy can be improved by considering the class labels of linked pages in addition to just the contents of the pages.

Recent researchers have developed many other approaches. Relational probability trees[18] is a complex learning algorithm taking the relations into consideration using probability trees[22], relational dependency networks[17] is also widely used. Under the topic of paper category classification, several approaches considering relations have been tried, some of the most representative ones are: [15] using Contribution-Based Cooperative Co-evolution(CBCC) algorithm, [5] using ReIF algorithm, [17]using relational probability tree, etc.

Pairwise Markov random field[25] is a widely used model in this area and [24] gives a broad introduction of this approach in network data. Similar works applying Markov

random field contains campaign detection[12], researcher relations[26], etc.

However, the above methods consider only one kind of relation. Therefore, ensemble methods are incorporated to combine different relations together. Voting is a simple and straightforward strategy. And a common way of voting strategy is weighted voting, originates from the weighted averaging ensemble method[13][20]. In the voting process, the ensemble algorithm chooses the most possible result according to the information gathered from individual learners. Another famous ensemble method is boosting, including AdaBoost[6] and XgBoost[2]. This group of algorithms aim at improving several weak learners into a strong one.

Ensemble method in relational classification is represented by [7] using ensemble classification for hyperlink classification. In that paper, ensemble classification methods have been incorporated to combine the results of the predictions for each hyperlink of a target page. [21] is a later work related to solving the paper category classification problem and shows strong performance, in this paper, several versions of ensemble methods are tried on the basis of collective classification.

3. METHOD

In this section, the idea and main components of our approach are presented in detail, containing four parts: problem formulation, local attributes, network relations and ensemble. The implementation details are covered in the experiment section(Sec 4).

3.1 Problem Formulation

The whole paper network can be represented by a graph G. G has V vertices denoted by $\{V_i|i = 1...V\}$, corresponding to papers with ID from 1 to V, and the label of vertice V_i is v_i. The local attributes of node V_i is the title of the paper, denoted by t_i, which is a vector in the form of bag of words.

Before constructing the edges in G, pairwise assumption is taken, which means that two vertices have relation with each other if and only if there is an edge between them. There are many possible relations in this problem, *e.g.* co-author, citation, they are denoted by $\{r_i|i = 1...R\}$, where R is the number of possible relations. Therefore, for an arbitrary pair of nodes V_i and V_j, there is an undirected edge between them if and only if they have some relation r_t, then the edge is denoted by e_{ij}^t. As for the factors of an edge e_{ij}^t, if relation t is a symmetric relation like co-author, then the factor $\phi_t(\cdot)$ is symmetric, *i.e.* $\phi_t(i, j) = \phi_t(j, i)$. On the contrary, if relation t is asymmetric, *e.g.* citation, then the factor $\phi_t(\cdot)$ is asymmetric, *i.e.* $\phi_t(i, j) \neq \phi_t(j, i)$.

After constructing all the connections between the nodes in G, we can split G into R sub-graphs, each containing all the vertices and edges belonging to the same relation. Theses sub-graphs are denoted by $\{G_i|i = 1...R\}$, G_i contains only the edges of relation i. For each kind of relation i, we will use the Markov Random Field over G_i correspondingly.

3.2 Local Attributes

As is mentioned in the formulation section, the local attributes of vertices in this problem contains the tile of paper only, and the title of paper i is denoted by t_i. Classify the paper with their titles is a classical problem in text classification. Many methods have been proposed in this area. However, as the title is too short to offer sufficient information, algorithms much too complex are wasteful and easy to overfit. Therefore, we use Naive Bayes for convenience and simplicity.

During training, we input the titles in the form of bag of words and the labels of every paper. After that, we will get the estimation of the parameters for Naive Bayes. In the classification phase, for papers in the test set, we input their titles into the algorithm and inference the categories of the paper using the parameters we estimate before. The output of our program is a 10-dimensional vector, representing the likelihood of each paper being in a certain category.

By using Naive Bayes, we can get a prior knowledge and approximate distribution (not very accurate, though) about the categories papers belonging to. Moreover, Naive Bayes is also a very good baseline. Later, we will combine the results from Naive Bayes with the relational classification in the ensemble part.

3.3 Network Relations

In this section, we present the approach for classifying papers with single relation network structure. More implementation details are covered in the experiment section.

3.3.1 Conditional Random Field

Conditional random field is a widely used tool for processing relational data and it is very similar to Markov random field. It composes of the graph G_i and the factors in it. We cover two steps in this part, the first is learning the value of the factors from the vertices in training set, the second is inferring the labels of the vertices that are hidden.

In our paper, we only consider pairwise situation. Therefore, the likelihood of observing certain labels of the vertices in graph $G_T$is

$$P(v_1, v_2, ..., v_n|\theta, x) = \frac{1}{Z(\theta, x)} \prod_{(V_i,V_j)\in E(G_t)} \phi_t(v_i, v_j)$$

where n is the number of vertices with observed labels, $E(G_T)$ denotes the set of edges in G_t, θ denotes the parameters *i.e.*the value of the factors, $Z(\theta)$ is the normalization term and x are the vertices we already know. Specifically, if we don't know any labels at all, the likelihood of observing certain configuration is

$$P(v_1, v_2, ..., v_n|\theta) = \frac{1}{Z(\theta)} \prod_{(V_i,V_j)\in E(G_t)} \phi_t(v_i, v_j)$$

3.3.2 Learning

In the learning phase, our goal is to estimate the value of the parameters θ, *i.e.* the value of the factors, in every graph G_t. The training set consists of n vertices with labels, denoted by $\{v_i|i = 1...n\}$. The ideal estimation of parameters θ has to maximize the likelihood of observing these n configurations. As the vertices outside the training set is hidden, conditional random field is the same as Markov random field in this step. That is

$$\theta = \arg\max_{\theta} P(v_1, v_2, ..., v_n|\theta)$$

which is also equivalent to

$$\theta = \arg\max_{\theta} \frac{1}{Z(\theta)} \prod_{(V_i,V_j)\in E(G_t)} \phi_t(v_i, v_j)$$

This model is able to derive good results for the following two reasons:

(1) This optimization problem is concave, as we can rewrite all the factors in a exponential form

$$P(v_1, v_2, ..., v_n|\theta) = \frac{1}{Z(\theta)} \prod_{(V_i,V_j)\in E(G_t)} \exp\left(\log \phi_t(v_i, v_j)\right)$$

After that we can turn to estimate the log form of the factor $\log phi_t(v_i, v_j)$ and solve it with many classical convex optimization algorithms like gradient descent and it is guaranteed to converge and reach the global optima of the function.

(2) The possible number of factors is small compared to the vertices in the training set. Take CORA as an example: there are only 10 possible categories in the data, which means the number of factors for one relation is at most 200 (take asymmetry into consideration); on the contrary, the number of the vertices is over 10000. This means that our estimation of the parameters is based on sufficient amount of data, thus being unlikely to overfit.

3.3.3 Inference

In the inference phase of relational data, the labels of the papers are classified based on the learned parameters and the papers with labels already known. The vertices with known labels are represented by $\{V_i|i = 1...n\}$, which is the same as above. The vertices to be classified is represented by $\{V_i|i = n + 1...V\}$. For each possible configuration for $V_{n+1}...V_V$, *i.e.* $v_{n+1}...v_V$, the likelihood of this configuration is

$$P(v_{n+1}, v_{n+2}, ..., v_V|\theta, v_1...v_n)$$

Therefore, the optimization function is

$$[v_{n+1}, ..., v_V] = \arg\max_{v_{n+1}\cdots v_V} P(v_{n+1}, v_{n+2}, ..., v_V|\theta, v_1, v_2, ..., v_n)$$

That is, finding the configuration with the greatest likelihood. During inference, enumerating all the possibilities with brutal force is time consuming. Instead, we use the **Belief Propagation Algorithm** in this part.

3.4 Ensemble

Ensemble classification completes the classification task by combining individual classifiers, thus achieves better performance than its components. In the former sections, the individual classifiers have been constructed, corresponding to the local attributes and different relations. Taking the example in the CORA data set, the classifiers deal with the titles, citation relationship and co-author relationship have to be combined. In this section, the individual classifiers are denoted by $\{h_i|i = 1...H\}$, and the possible labels are $\{c_i|i = 1...C\}$.

3.4.1 Voting

Voting is a widely used algorithm in ensemble learning for its simplicity and practicality. We implement the voting strategy for our ensemble learning. Voting tends to have better performance when the diversity of the individual classifiers is large. The main difficulty in our algorithm is that, the number of individual classifiers is small (3 in CORA), while there are ten categories in all. But as the classifier generated from relations and local attributes are heuristically different, the performance of the classifiers can have a slight increase with reasonable voting strategy.

Our voting strategy is as follows. Considering a node V in the test set, **(1)**If more than half of the individual classifiers give the same judgment, then it is the answer. In the paper classification problem, as there are only 3 individual classifiers, two classifiers giving the same result will be considered persuasive; **(2)**If the individual classifiers cannot achieve agreement, then we further gathering their judgment. Every individual classifier h_i gives its classification result in the form of a C-dimensional vector s_i, in which s_{ij} denotes the likelihood of label j in the classification of classifier h_i. After this, the weighted average is implemented on vectors $s_1...s_H$, and then the best classification is got.

3.4.2 Learning

Learning method learns a strategy to combine all the classifiers if the data is sufficient. In this paper, we use a shallow neural network to learn the mapping from the results of individual classifiers to a more accurate classification. The main advantage of this method is that, the neural network can implicitly learn the complex links between the features.The network takes the likelihood of each label from different classifiers as input. And the output is the likelihood of the paper belonging to each category. The main difficulty of the neural network is the easiness to overfit, and that is the reason of using a shallow neural network instead of a deep one. Due to the constrain of the time, its difficult to train our network to converge, thus we are unable to experiment fully in this approach. But the neural network still turns out to function well on our task.

4. EXPERIMENT

4.1 Implementation Details

4.1.1 Preprocessing

In preprocessing phase, the data set is split into training set and test set in the proposition of 10%, 25% and 40%, each possible proposition have ten possible splits correspondingly. The simplest and most straight-forward way is randomly choosing test set papers from the whole data set. However, to best leverage the network structure, we initially planned to use the snowball sampling algorithm[10]. But we were only able to implement random selection algorithm within deadline.

4.1.2 Local Attributes

The local attributes in the CORA data set is the titles of the papers. First the papers without a known label are omitted in both training set and test set, and the titles of the papers are represented in the form of bag of words. After that, Naive Bayes algorithm is applied to the new data set in C++. Its output is the possibility distribution over ten categories according to its inference.

4.1.3 Network Relations

The conditional random field in this paper is implemented with the basis of UGM toolkit[23] in Matlab. With the help of this toolkit, the learning and inference can be efficiently done after transforming the paper network in a specific form of matrix. The output of this algorithm is the predicted label of a paper and the marginal distribution of this paper belonging to each category.

Several other issues are very tricky and worths mentioning:

1. **Optimization:** We tried several versions before we reached a normal set of results. The main problem lies in the optimization. The default optimization and initialization method is SGD and set all factors to zero, which is quite time consuming and hard to converge. Later we tried the Qausi-Newton method and initialize the factors according to Gaussian Distribution, and this time the results turn out to be much better.

2. **Asymmetric factors** The UGM only models the undirected graph, therefore we have to use the undirected graph to represent the directed relations. The solution is as follows. We define three kinds of factors $\phi^{(1)},\phi^{(2)},\phi^{(3)}$. For asymmetric factors$\phi(i,j)$, if$i > j$, then $\phi(i,j) = \phi^{(1)}(i,j) + \phi^{(2)}(i,j)$, if$i < j$, then $\phi(i,j) = \phi^{(1)}(i,j) + \phi^{(3)}(i,j)$. In this way, we can differ asymmetric from symmetric factors.

3. **Co-author relations**. If we simply connects the vertices with the same author, then there is a clique for each one of the authors, then the complexity of the networks will grow greatly. In our implementation, the papers with the same author are arranged in a chain, thus decrease the density and complexity of the graph.

4.1.4 Ensemble

Two types of ensemble methods are tried in our approach.

Voting strategy in our approach is simply weighted average the probability distribution from different classifiers. The weights of distribution from author relation classifier, citation classifier and Naive Bayes classifier are 0.1, 10 and 0.4. As the weights of voting can easily be influenced by the training set and overfit, we donot get the weights by learning. This set of weights is achieved by doing experiments on the validation set.

We implement a shallow neural network to combine the information from different classifiers with pytorch[19]. The network takes probability distribution from citation classifier and local attribute classifier as input. The network has 3 layers, each with 20, 15 and 10 neurons. The neurons are all linear units with all the activation functions being RELU. In the last layer, we use a softmax layer with a width of ten, mapping our results to a probability distribution over ten categories. Finally, we select the category with the highest probability from the output. During training, we select 60% of data from the network as training set and keep the model fixed for further test.

4.2 Evaluation Metric

CORA data set is a classification problem with ten labels. A common way to evaluate the performance of the algorithms in such problems is considering top-k accuracy. However, most of the previous approaches simply use accuracy as the evaluation metric, including the state-of-art approach. Therefore, we accordingly use accuracy in our experiment.

4.3 Performance

The experiment covers two aspects of comparison: performance among approaches and performance under different proportion of training set.

Table 1: Comparison of accuracy

Algorithm	Accuracy
CBCC[15]	0.754
RPT[17]	0.792
MLN[3]	0.798
EPRN[21]	0.840
ReIF[5]	0.857*
CRF-Voting	0.771
CRF-NN(40%)	0.917
CRF-NN(25%)	0.935
CRF-NN(10%)	0.825

Table 2: Comparison of accuracy

Accuracy / Proportion / Algorithm	10%	25%	40%
Naive Bayes	0.702	0.691	0.652
CRF	0.752	0.722	0.657
CRF-Voting	0.770	0.740	0.693
CRF-NN	0.825	0.935	0.917

4.3.1 Comparison of different approaches

In the first part, we compare our approach with previous ones. To the best of our knowledge, previous approaches contains [15][5][17][3][21]. Our algorithms are CRF-Voting and CRF-NN, representing the ensemble methods voting and neural network. The accuracy of previous methods are tested under 25% or 10% percent of test set. So we set 25% of the data as test set. The accuracy is got after running on ten different splits of the data set and averaging. The result is shown in **Table 1**. It can be noticed that using neural network improves the performance greatly, even outperforms the best previous algorithm under 25% test set. However, the reason for the result of neural network being inconsistent remains unknown, we made a conjunction in the following part. In the second part, we discuss the influence of the proportion of test set.

4.3.2 Comparison of different proportions of test set

In this part, we compare the performance of different algorithms under several proportions of test set. With the time constrained, we are only able to do the experiments on 10%, 25% and 40%. We compare the performance of Naive Bayes, Markov random field with asymmetric factors and the final combination with ensemble methods. The result is shown in **Table 2**. From the figure, we notice that using relations is slightly better than using local attribute only. If we apply ensemble to them, the performance will improve, even the ensemble method is as simple as a weighted average voting.

In this problem, neural network shows its strong power to fit an unknown non-linear function, improves the performance by a large margin. However, it suffers from the problem of inconsistency. Considering that our model is trained with 60% percent of data in training set, 25% of test set is classified with more known data, thus increase the information and improve the accuracy. However, as the test set takes up only 10%, the graph of only test set becomes more sparse and the classification difficulty is increased. However, we are still to see our algorithm gives a not bad result.

5. DISCUSSION

5.1 Factor Asymmetry

The paper category inference problem in CORA data set consists of both symmetric and asymmetric relations. The symmetric relations are co-author relations and the asymmetric relations are citation relations. In our experiment, we observed that taking the direction of relations into consideration improves the performance of the algorithm. When the proportion of test set is 25%, the accuracy of undirected relationship CRF is 66%, while the accuracy of directed relationship CRF is 72%. Heuristically, this is because of the additional information from the direction. After considering the direction, the number of the factors is doubled, and the direction of belief propagation is constrained and changed. These may all lead to the improvement of accuracy.

5.2 Further Improvement

As our time is strictly constrained this time, much future work can be done to improve the performance.

The first is the modeling of the structure. In many previous algorithms, the classification problem is solved by collective classification, *i.e* they only deal with the neighbourhood structures in a network, using algorithm like ICF and GS. Theoretically, MRF and CRF are able to achieve the same result, or they can be even better as the global conditions of the network are considered in their modeling. However, with the growing number of vertices in the network, the complexity and size of MRF and CRF also increase greatly. This may lead to the precision loss during inference and learning and resulting in an accuracy debasement. In the future, better implementation for MRF and CRF will improve the performance. Moreover, combining collective classification with MRF and CRF is also a good choice, as we can collect their advantages and achieve better performance.

The second is the ensemble method. Ensemble is able to get a better result only when the original accuracy and the diversity of the individual classifiers are both promised. However, in this paper, the number of the classifiers is only three and the best of them can only get an accuracy of 72%(CRF). Therefore, more classifiers and highly-performing classifiers is a must for the improvement of current algorithms on this sort of problems

6. CONCLUSION

In this paper, we present our CRF-Ensemble approach for classification in network using relational data. Our method contains three phases,each dealing with local attributes, network relations and ensemble. In the evaluation part, we show that probability graphical model and ensemble can achieve much better performance than using local attributes only. Moreover, we discuss some detailed topics on how to improve the classification accuracy.

7. ACKNOWLEDGMENT

First, we want to thank Prof.Yizhou Sun and Prof.Yongzhi Cao for this excellent course. Second, we also want to thank TA for his hard work. Third, We believe we have to thank Google, Baidu, Github, overleaf, wechat and mail.pku.edu.cn for techinical support. Finally, we thank Peking university for the place and the staff of drama "West World" for offering entertainment.

8. REFERENCES

[1] S. Chakrabarti, B. E. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks, 2002.

[2] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

[3] R. Crane and L. McDowell. Investigating markov logic networks for collective classification. In *ICAART (1)*, pages 5–15, 2012.

[4] T. G. Dietterich. Ensemble methods in machine learning. *Proc International Workshgp on Multiple Classifier Systems*, 1857(1):1–15, 2000.

[5] Q.-T. Dinh, C. Vrain, and M. Exbrayat. A link-based method for propositionalization. In *ILP (Late Breaking Papers)*, pages 10–25. Citeseer, 2012.

[6] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[7] J. FÃijrnkranz. Hyperlink ensembles: a case study in hypertext classification. *Information Fusion*, 3(4):299–312, 2001.

[8] L. Getoor and B. Taskar. An introduction to conditional random fields for relational learning. *Foundations TrendsÂő in Machine Learning*, 4(4):267–373, 2010.

[9] W. R. Gilks, S. Richardson, and D. Spiegelhalter. *Markov chain Monte Carlo in practice*. CRC press, 1995.

[10] L. A. Goodman. Snowball sampling. *Annals of Mathematical Statistics*, 32(1):148–170, 1961.

[11] D. Jensen, J. Neville, and B. Gallagher. Why collective inference improves relational classification. In *Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 593–598, 2004.

[12] H. Li, A. Mukherjee, B. Liu, R. Kornfield, and S. Emery. Detecting campaign promoters on twitter using markov random fields. In *IEEE International Conference on Data Mining*, pages 290–299, 2015.

[13] H. Markowitz. Portfolio selection. *Journal of Finance*, 7(1):77–91, 1952.

[14] A. K. Mccallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

[15] L. K. McDowell, K. M. Gupta, and D. W. Aha. Case-based collective classification. In *FLAIRS Conference*, pages 399–404, 2007.

[16] J. Neville and D. Jensen. J. neville and d. jensen (2000). iterative classification in relational data. proceedings of the aaai 2000 workshop learning statistical. 2000.

[17] J. Neville and D. Jensen. Collective classification with relational dependency networks. In *Proceedings of the Second International Workshop on Multi-Relational Data Mining*, pages 77–91. Citeseer, 2003.

[18] J. Neville, D. Jensen, L. Friedland, and M. Hay. Learning relational probability trees. In *ACM SIGKDD International Conference on Knowledge*

Discovery and Data Mining, pages 625–630, 2003.

[19] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[20] M. P. Perrone and L. N. Cooper. *When networks disagree: Ensemble methods for hybrid neural networks*. 2015.

[21] C. Preisach and L. Schmidt-Thieme. Relational ensemble classification. In *null*, pages 499–509. IEEE, 2006.

[22] F. Provost. Well-trained pets: Improving probability estimation trees. 2000.

[23] M. Schmidt. Ugm: A matlab toolbox for probabilistic undirected graphical models. http://www.cs.ubc.ca/ schmidtm/Software/UGM.html, 2007.

[24] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93, 2008.

[25] B. Taskar, M. F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *International Conference on Neural Information Processing Systems*, pages 659–666, 2003.

[26] C. Wang, J. Han, Y. Jia, J. Tang, D. Zhang, Y. Yu, and J. Guo. Mining advisor-advisee relationships from research publication networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, Dc, Usa, July*, pages 203–212, 2010.