Mining Research Topic-Related Influence between Academia and Industry

Dan He

Computer Science Dept., Univ. of California, Los Angeles, CA, 90095-1596, USA
danhe@cs.ucla.edu

Abstract. Recently the problem of mining social influence has attracted lots of attention. Given a social network, researchers are interested in problems such as how influence, ideas, information propagate in the network. Similar problems have been proposed on co-authorship networks where the goal is to differentiate the social influences on research topic level and quantify the strength of the influence. In this work, we are interested in the problem of mining topic-specific influence between academia and industry. More specifically, given a co-authorship network, we want to identify which academia researcher is most influential to a given company on specific research topics. Given pairwise influences between researchers, we propose three models (simple additive model, weighted additive model and clustering-based additive model) to evaluate how influential a researcher is to a company. Finally, we illustrate the effectiveness of these three models on real large data set as well as on simulated data set.

1 Introduction

In recent years, the problem of mining influence in networks, especially social networks, has attracted tremendous attention [5] [7] [9] [10]. In traditional social network, nodes are usually individuals and edges indicate friendship between the pair of individuals. One of the key questions in social network is how ideas, information or influence spreads (cascades) through the network.

Different to traditional social network, in co-authorship network, nodes are researchers and edges indicate the co-author relationship of the pair of researchers. There are weights associated with the edges, indicating the number of publications co-authored by the pair of researchers. Another important difference is there are events, or actions, associated with individuals in the traditional social network. These actions are usually temporal, namely each action has a specific occurrence time. For example, if the action is "purchase ipad", a person and his/her friends in the network may take the action at different times. This temporal property allows quantification of the influences between different individuals and the study of the influence spread model in the traditional social network. The co-authorship network, on the contrary, usually do not have the temporal property. A researcher always publishes papers the same time with his co-authors. Therefore, the influence in the co-authorship network should be defined differently.

In this work, we are specifically interested in the topic-level influence between academia and industry. We believe this is an important problem in that it helps people to evaluate which academia researcher has better connection to a company, and vice

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part II, LNAI 6912, pp. 17–31, 2011.

versa. This can be very useful in many cases. For example, when a student is seeking an advisor and his career goal is to be a researcher in a company's research lab, he may want to choose an advisor who has tight connection to the company. Another example is funding agencies may choose to award certain type of grants to researchers who work closely with industry companies. Companies may also want to collaborate with academia researchers who have tight industry connection in the same fields. To our knowledge, there is no prior work on mining influence between academia and industry.

A model using Topical Affinity Propagation (TAP) to learn the topic-level social influence on large networks has been proposed recently [13]. Based on the topic-level influence identified by TAP, we proposed three models to mine the topic-level influence of a researcher to a company: (1) simple additive model where we simply sum the influence of the researcher to all the researchers in the company. (2) weighted additive model where we weight the researchers in the company by their internal influence in the company. (3) clustering-based additive model where the researchers in the company are clustered first and then each cluster of closely related researchers (who often publish together) is considered as a "super researcher". We then evaluate the three models on real co-authorship network as well as on simulated co-authorship networks.

2 Related Work

There have been lots of work recently on the problem of mining influence in networks, especially social networks. These work are mainly focused on two main categories of problems: influence probabilities between nodes in the network are pre-defined or these probabilities need to be learned.

For the first category, Domingos and Bichardson [5] studied the viral marketing problem, which targets the most influential users in the network, by viewing the market as a social network and modeled it as a Markov random field. Kempe et al. [7] studied the influence maximization problem, which selects an initial set of users who eventually influence the largest number of users in the social network. A greedy algorithm as well as the first provable approximation guarantees for efficient algorithms are provided. Leskovec et al. [9] modeled the outbreak detection problem as selecting nodes in a network in order to detect the spreading of a virus or information as quickly as possible. Rodriguez et al. [10] developed an approximation algorithm for the problem of identifying optimal diffusion network from temporal data. The algorithm is able to be scaled to large network to trace paths of diffusion and influence through networks and to infer the optimal network that best explains the influence propagation. Chen et al. [4] develop methods to improve the efficiency of the greedy algorithm in [7] to maximize influence. New degree discount heuristics that improve influence spread are further proposed.

For the second category, Goyal et al. [6], studied the problem of learning influence probabilities from historical user action data and try to predict when the users will get activated from the learned probabilities. Saito et al. [11] applied EM algorithm to solve the same problem focusing on the Independent Cascade model of propagation.

Tang et al. [13] introduced a topic-specific social influence problem. Instead of friends, the nodes in the networks are co-authors of one another. Each researcher of the network is associated with a distribution of topics, which are the research topics the

researcher had publication in. Topic models [12] are applied to automatically extract topics from the publications and to initialize the topic distribution of each node. Given a co-authorship network and the topic distribution of the nodes, a topical factor graph [8] is built, in which the observation data are cohesive on both local attributes and relationships. The nodes and edges in the co-authorship network represent the observation data and the relationship in the factor graph, respectively. Three kinds of feature functions are proposed: (1) Node feature function which measures the similarity of the nodes based on their topical similarity or topical interaction strength (using co-authorship information); (2) Edge feature function which measures if there is an edge between the two nodes in the network; (3) Global feature function which measures the representative node on a specific topic. A joint likelihood function is then proposed as the product of the three feature functions for all the nodes in the graph. A Topical Affinity Propagation (TAP) on the factor graph is designed to maximize the likelihood function. The topic specific influence from node a to node b is then defined using a sigmoid function based on two messages in the propagation: how likely node a thinks it influences node b and how likely node b thinks it is influenced by node a on the topic. Therefore, the influence between two nodes are usually not symmetric. Two different propagation frameworks were proposed: a message passing framework and a Map-Reduce framework. We refer to the paper [13] for the details of the model.

3 Models

Since the focus of this work is to model the influence between academia and industry in stead of influence between individual researchers, we first assume we already identify pairwise influences between researchers in the co-authorship network, where the influence needs to be a value of real number. There are multiple methods to identify pairwise influences between researchers. Maybe the most naive method is based on the number of co-authored papers. The influence from author A to author B can be defined as the percentage of their co-authored papers in all the papers published by B. However, this naive method considers a pair of authors as independent to other authors, which is usually not the case. On the contrary, the TAP method [13] applies affinity propagation and therefore is able to better model influences between multiple researchers. In this work, we take the TAP method to estimate the pairwise influences, which are within t the range of [0, 1]. The influences are directed and therefore usually not symmetric. Next we discuss three different models to mine influence from an academia researcher to a company. The influence from industry to academia can be obtained using the same models but with reverse influence direction.

3.1 Simple Additive Model

In this model, we simply sum the topic-specific influence from a researcher to all the researchers in a company. Then we take the sum as the topic-specific influence between the researcher and the company. Therefore, the topic-specific influence from a researcher to a company under *Simple Additive Model* is defined as the following:

$$I_t(r, C) = \sum_{i=1}^{n} I_t(r, C_i) \tag{1}$$

where r is the researcher, C is the company, $I_t(r, C)$ is the influence from r to C for topic t, C_i is the i-th researcher in company C, $I_t(r, C_i)$ is the influence from r to C_i for topic t. Naturally the influence from an university to a company $I_t(U, C)$ is defined as the following:

$$I_t(U, C) = \sum_{i=1}^{n} I_t(U_i, C) \tag{2}$$

where $I_t(U_i, C)$ is the influence from the i-th researcher in university U to the company C. Therefore, the influence from an university to a company is simply the sum of the influence from every researcher in the university to the company.

3.2 Weighted Additive Model

The simple additive model has a problem that all the researchers in the company are weighted equally. This is usually not the case. For example, a manager should have a higher influence in the company than a junior researcher. Therefore, the same influence to the manager and to the junior researcher should not mean equal influence to the company.

To address the above problem, we first compute the internal influence for the researchers in a company. Then each researcher is weighted according to their internal influence. The weights are applied to the researchers when we sum the influences to them. The *Weighted Additive Model* is defined as the following:

$$W_t(C_i) = \frac{\sum_{j=1}^{n} I_t(C_i, C_j)}{\sum_{j=1,k=1}^{j=n,k=n} I_t(C_k, C_j)} \tag{3}$$

$$I_t(r, C) = \sum_{i=1}^{n} I_t(r, C_i) \times W_t(C_i) \tag{4}$$

Where $W_t(C_i)$ is the weight for the i-th researcher in company C for topic t. Similarly, the influence from an university to a company can be computed via Equation 2.

3.3 Clustering-Based Additive Model

The weighted additive model may still have a problem. We show the problem by an example in Figure 1. In this example, A, B are academia researchers. C, D, F, E, G, H are researchers in the same company. Let's assume C, D always publish paper together. Therefore the influences between C, D are relatively high. E influences F, G, H but the influence is lower than the influence between C, D. Therefore, it's possible that the weights of C, D are higher than the weight of E. Thus even though A influences C, D the same extent as B influences E, higher weight of C, D makes A more influential to the company.

However, it may be the case that one of C, D is a senior researcher and the other is a junior researcher. On the contrary, all the researchers E, F, G, H are senior. Therefore, although E has lower influence to F, G, H, E should naturally have a higher influence

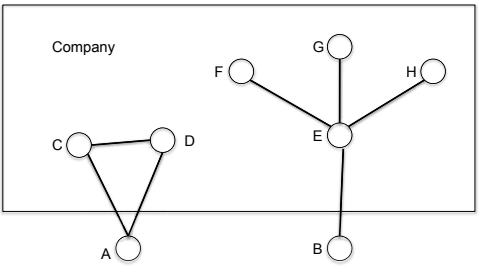

Fig. 1. Example to illustrate the potential issue of weighted additive model

in the company. Thus to determine the weight of a researcher, we may not simply sum the influence of the researcher to others in the same company. Instead, we should consider how the researchers in the company correlates with each other, or how often they publish paper together. By using the correlations, we can cluster the researchers that collaborate a lot into clusters. We then consider each cluster as a *super researcher*. We average the influence in a cluster. Then we can apply the weighted additive model on the clusters instead of on researchers. We call the model *Clustering-based Additive Model*. The benefit of the clustering-based additive model is we address the correlation between researchers in the same company such that the weights on more correlated researchers can be adjusted appropriately.

There are many existing clustering algorithms such as K-means, EM etc.. The problem of these existing clustering algorithms is that the number of clusters is not known and different numbers affect the results of the clustering algorithm. Methods such as modularity based clustering [3] are able to figure out the most likely cluster numbers automatically. However, modularity based clustering only considers the number of nodes in the cluster. It does not consider the distances between the nodes. In our co-authorship network, besides the information that whether the authors collaborate with each other, we also care about how much they collaborate. Therefore the distances between the nodes are important information and should not be ignored. We next present a new method that takes into account the distances between the nodes.

In order to avoid choosing a cluster number, we can build a graph $G = (V, E)$ where each researcher is a node and $e_{i,j} \in E$ iff the similarity between node i and j, $s_{i,j} \geq t$ and t is a similarity threshold. $s_{i,j}$ is defined as the Jaccard's coefficient of the publications of researcher i and j, namely:

$$s_{i,j} = \frac{|P(i) \cap P(j)|}{|P(i)| + |P(j)| - |P(i) \cap P(j)|} \tag{5}$$

where $P(i)$ is the set of publications of researcher i, $|P(i)|$ is the number of publications for set $P(i)$, $P(i) \cap P(j)$ is the set of publications co-authored by researchers i and j. We use co-authorship instead of influence to define the similarity since the co-authorship is symmetric while the influence is directed and not symmetric.

Once we have the graph, we can cluster the nodes in the graph such that the nodes in the same cluster are fully-connected, namely each cluster is a clique. We can apply clique searching algorithms to easily find cliques. However, one problem is a node may

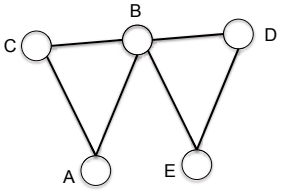

Fig. 2. Example to illustrate a node B is shared by two cliques $\{A, B, C\}$ and $\{B, E, D\}$

Input: A Graph $G = (V, E)$ and a threshold t
Output: A set of clusters $cluster^t$
1. While (there are nodes to be selected)
2. Randomly select a node a
3. Remove a and all the edges associated with a from G
4. assignCluster(a)
5. End
6. Output all the clusters

Fig. 3. Algorithm to generate clusters from a graph given a similarity threshold t

be shared by two possible cliques, as shown in Figure 2. We take a greedy strategy that during the clique searching process, once we assign a node to a clique, we remove the node as well as all the edges connected to the node from the graph. Thus given a threshold t, we can generate a set of clusters $cluster^t$. The algorithm to generate a set of clusters given a threshold t is shown in Figure 3. In line 4, the function assignCluster(node) assigns the node to an existing cluster such that the resulting cluster is still a clique. If there is no such existing cluster, we create a new cluster and assign the node to the new cluster. Since assigning the node to a cluster requires checking the distance of the node to all the nodes in the cluster, the time complexity of assignCluster(node) is $O(n)$. Therefore the complexity of the algorithm is $O(n^2)$ because we need to assign cluster for all the n nodes.

Next we want to determine a good threshold t automatically. We first define an objective distance function as the following:

$$F(t) = || \sum_{k=1}^{k=|cluster^t|} \sum_{i,j \in cluster_k^t} (1 - s_{i,j}) - |cluster^t|| | \tag{6}$$

where t is a similarity threshold, $cluster^t$ is the set of clusters when using threshold t, $cluster_k^t$ is the k-th cluster, $s_{i,j}$ is the similarity between nodes i, j, $|cluster^t|$ is the number of clusters. We use $1 - s_{i,j}$ instead of $s_{i,j}$ directly since the more similar two nodes are, the shorter the distance between the two nodes.

The objective function is based on the motivation that we want to minimize the distances of the nodes within the same cluster. Therefore nodes far from each other won't be put in the same cluster. But since obviously the more clusters, or the smaller number of nodes each cluster has, the smaller the distances of the nodes within the same cluster.

If we do not consider the number of clusters, then t=1 will always minimize $F(t)$ since with t=1, the graph contains no edges (we assume the similarity of any pair of nodes is less than 1 and all the nodes will by themselves be a cluster. Thus $F(t)$=0.Therefore, we need to take into consideration the number of clusters. This is similar to how people select a model using AIC or BIC scores [2]. To select a best model, besides minimizing the objective function, the number of the parameters in the model is also taken into account such that the model with smaller parameters is preferred. We noticed that similar ideas have been used in the work of [14].

To find a threshold t to minimize $F(t)$, an extensive search is not applicable since t is continuous. However, in our problem, the number of researchers, or nodes, in the graph is finite. We can compute the pairwise similarity of the nodes in the graph as $s_{1,2}, s_{1,3}, \ldots, s_{n,1}, s_{n,2}, \ldots, s_{n,n-1}$ where n is the total number of nodes. Thus we can try t as $s_{i,j}$ for all i's, j's such that $1 \leq i, j \leq n$ and $i \neq j$ and we show at least one of the $t = s_{i,j}$'s minimizes $F(t)$.

Lemma 1. *At least one of the $s_{i,j}$'s for $1 \leq i, j \leq n$ is able to minimize $F(t)$.*

Proof: We can sort the n^2 $s_{i,j}$'s in ascending order to $s_1, \ldots, s_{n^2}$ then we have $s_k \leq s_{k+1}$ for all $1 \leq k \leq n^2$. For all $s_k < t < s_{k+1}$, the edges in the graph will be exactly the same as the edges in the graph where t=s_k. Therefore, the set of clusters when $s_k < t < s_{k+1}$ will also be identical to the set of clusters when t=s_k and $F(t)$ for both cases are also identical. Thus for any given t, there will be a corresponding s_k such that $s_k < t < s_{k+1}$ and $F(t)$=$F(s_k)$. Thus we can conclude that at least one of the $s_{i,j}$'s for $1 \leq i, j \leq n$ is able to minimize the function $F(t)$. ■

Therefore, the number of trials for t is $O(n^2)$. The overall complexity of our method is thus $O(n^4)$ since for each trial of t, we need to run the algorithm shown in Figure 3 to obtain clusters, whose complexity is $O(n^2)$.

The complexity of our method seems very high, but in reality, n for a company is usually small. The number of trials can be smaller than n^2 as well since the similarity of different pair of nodes can be the same and we only need to try unique values of t. Also most researchers only have a small number of collaborators. If we assume a researcher has constant number $h << n$ of collaborators, instead of $O(n^2)$ edges, the graph has $O(n \times h)$ edges. Therefore the total complexity of the algorithm is $O(n^2 \times h^2)$. Since h is constant, the complexity approximates to $O(n^2)$. So enumerating all $s_{i,j}$'s is not a problem. In our experiments on real data set, the running time is actually in seconds. Once we find a threshold to minimize $F(t)$, we obtain a set of clusters using the algorithm shown in Figure 3. Then to compute the influence of a researcher to a company, we deploy a hierarchical framework.

We first compute the weight of each researcher in every cluster as the following

$$W_t(L_i) = \frac{\sum_{j=1}^{n} I_t(L_i, L_j)}{\sum_{j=1,k=1}^{j=n,k=n} I_t(L_k, L_j)} \tag{7}$$

where L_i is the i-th researcher in the cluster L on topic t, $I_t(L_i, L_j)$ is the influence between L_i and L_j on topic t. We then compute the influence between clusters L and K, $I_t(L, K)$ in the same company on topic t as the following:

$$I_t(L,K) = \sum_{i=1,j=1}^{i=|L|,j=|K|} I_t(L_i,K_j) \times W_t(L_i) \times W_t(K_j)$$

where $W_t(L_i)$ and $W_t(K_j)$ are computed via the Equation 7, $I_t(L_i,K_j)$ is the influence between researchers L_i and K_j on topic t. As we can see, the influence between two clusters are weighted on the researchers in both clusters. We call the influence $I_t(L,K)$ as the *cluster-based* influence. The general procedure to cluster the researchers in a company and to comput the corresponding weights of the cluster in shown in Figure 4. Finally we consider each cluster as a super researcher where the weighted additive model can be applied directly. The influence of each academia researcher r to each cluster L in the company C is computed using the weighted additive model described previously. For each cluster we only consider the researchers in the same cluster and the influence $I_t(r,L)$ is computed in the following way:

$$I_t(r,L) = \sum_{i=1}^{n} I_t(r,L_i) \times W_t(L_i)$$

Thus we compute the influence from a research r in academia to a company C on topic t, $I_t(r,C)$ as the following:

$$W_t(C_L) = \frac{\sum_{K \in cluster(C)} I_t(C_L,C_K)}{\sum_{L \in cluster(C), K \in cluster(C)} I_t(C_L,C_K)}$$

$$I_t(r,C) = \sum_{L \in cluster(C)}^{n} I_t(r,C_L) \times W_t(C_L)$$

where $W_t(C_L)$ is the weight of the cluster L in company C computed via the cluster-based influence.

Input: A Graph $G = (V,E)$ for researcher in a company
Output: A set of clusters and their weights
1. Compute the optimal similarity threshold
2. Generate clusters given the threshold
3. For each cluster
4. compute the weights of the researchers in the cluster
5. End
6. Compute the weights of each cluster
7. Return the clusters and their corresponding weights

Fig. 4. Procedure to cluster the researchers in a company and to compute the weights of the clusters

4 Experimental Results

4.1 Experiments Settings

We use the same data set used by Tang et al. [13]. There are totally eight different topics. The number of researchers in each topic is shown in Table 1. Notice the same researcher may have multiple research areas. We also show the number of companies and universities working on the topics. Similarly, the same company or university may have multiple research areas. The numbers of companies and universities may not be accurate though since there are researchers with missing affiliation information. We obtain the pairwise topic-specific influence between researchers using the model in [13]. The influences are directed and therefore usually not symmetric. The affiliation information of the researchers is actually annotated in the data set of [13]. And each researcher has only one affiliation in the data set. We just use the researchers with affiliation annotation and ignore the others without such information. As we discussed before, most of the researchers should have much fewer collaborators than the total number of researchers. In the data set, on each topic, we observe the researchers influence on average less than 5 researchers from the same company. Therefore our clustering algorithm runs very fast and is able to finish in seconds.

Table 1. Topics in our data set and the number of corresponding researchers, companies, universities for each topic

Topics	#Researchers	#Companies	#Universities
Data Mining	679	33	99
Machine Learning	976	48	97
Database System	1127	66	116
Information Retrieval	657	49	87
Web Services	400	27	48
Semantic Web	671	38	35
Bayesian Network	554	24	45
Web Mining	348	25	47

4.2 Influence of Academia Researchers to Company

We first show the top-5 most influential researchers to the company 'Microsoft' and 'IBM' on 'Data Mining' under different models. The researchers are ranked by their influences under different models. As we can see, in Table 2, For IBM, the most influential researchers and their corresponding ranks are identical under the three models. This is due to the number of researchers in IBM on Data Mining in the collected data set is relatively small (5 in total). The models did change the influences of these researchers but the ranks of them still remain the same. On the contrary, the rank of the most influential researchers changed for Microsoft under these models. The reason the ranks of some researchers become lower from simple additive model to weighted additive model is because in our data set the researchers at Microsoft that were influenced by these academia researchers have relatively low influence in the company.

When the researchers in the company is clustered, the influence of the researchers who have high influence on the clustered researchers will be changed dramatically.

Table 2. The top-5 most influential researchers to the company Microsoft and IBM on data mining under different models. The researchers are ranked by their influences under different models.

Data Mining (Microsoft)			Data Mining (IBM)		
simple additive	*weighted additive*	*clustering-based additive*	*simple additive*	*weighted additive*	*clustering-based additive*
Jiawei Han	Huan Liu	Clark Glymour	Jiawei Han	Jiawei Han	Jiawei Han
Huan Liu	Clark Glymour	Huan Liu	Philip S. Yu	Philip S. Yu	Philip S. Yu
Xifeng Yan	Michail Vlachos	Michail Vlachos	Michail Vlachos	Michail Vlachos	Michail Vlachos
Clark Glymour	Xuanhui Wang	Padhraic Smyth	Tao Tao	Tao Tao	Tao Tao
Philip S. Yu	Padhraic Smyth	Bing Liu	Ricardo Vilalta	Ricardo Vilalta	Ricardo Vilalta

Table 3. The top-5 most influential researchers to the company Microsoft and IBM on database systems under different models. The researchers are ranked by their influences under different models.

Database Systems (Microsoft)		
simple additive	*weighted additive*	*clustering-based additive*
Calton Pu	Venkatesh Ganti	Sharad Mehrotra
Jiawei Han	Luis Gravano	Jiawei Han
Sharad Mehrotra	Sharad Mehrotra	Jeffrey F. Naughton
Venkatesh Ganti	Jeffrey F. Naughton	Venkatesh Ganti
Jeffrey F. Naughton	Jiawei Han	Luis Gravano
Database Systems (IBM)		
simple additive	*weighted additive*	*clustering-based additive*
Kevin Chen-Chuan Chang	Joseph M. Hellerstein	Joseph M. Hellerstein
Joseph M. Hellerstein	Kevin S. Beyer	Kevin S. Beyer
Renee J. Miller	Renee J. Miller	Min Wang
Kevin S. Beyer	Michael J. Franklin	Kevin Chen-Chuan Chang
Michael J. Franklin	Kevin Chen-Chuan Chang	Michael J. Franklin

For the researchers at Microsoft on data mining, we identified two researchers who publish together frequently. More specifically, one researcher published 39 papers, the other published 118 papers and they co-authored 32 papers. It might be the case that one researcher is relatively senior and the other researcher is relatively junior. We then further identified the title of the researcher that is suspected to be senior and he is indeed a manager of Microsoft. The two researchers are grouped in one cluster and therefore the weights of the two researchers are adjusted appropriately. This leads to the change of the influence for academia researchers who influence these two researchers.

One thing to notice is that our data set actually misses affiliation information for many researchers. Therefore, the rank of the most influential researcher on the companies may not be accurate. But the data set is big enough to show the effectiveness of our models and algorithms.

We also show the top-5 most influential researchers to the company 'Microsoft' and 'IBM' on 'Database Systems' and 'Machine Learning' under different models in Table 3 and 4, respectively. Similarly, for database systems researchers at Microsoft, we identify 4 clusters consist of one senior researcher and one junior researcher. We identified the title of those senior researchers and all of them are manager or principal/senior researcher. We show the number of publications by each of them and the number of their co-authored publications in Table 5. As we can see, the junior researchers published at least half or even 80% of their papers with the senior researchers. Therefore,

Table 4. The top-5 most influential researchers to the company Microsoft and IBM on machine learning under different models. The researchers are ranked by their influences under different models.

Machine Learning (Microsoft)		
simple additive	*weighted additive*	*clustering-based additive*
Brendan J. Frey	Aaron Hertzmann	Aaron Hertzmann
Aaron Hertzmann	Michael I. Jordan	Michael I. Jordan
Andrew McCallum	Andrew McCallum	Andrew McCallum
Michael I. Jordan	Brendan J. Frey	William T. Freeman
William T. Freeman	William T. Freeman	Yoav Freund
Machine Learning(IBM)		
simple additive	*weighted additive*	*clustering-based additive*
Inderjit S. Dhillon	Inderjit S. Dhillon	Inderjit S. Dhillon
Adam R. Klivans	Manfred K. Warmuth	Manfred K. Warmuth
Nader H. Bshouty	Roni Khardon	Roni Khardon
Joydeep Ghosh	Geoffrey J. Gordon	Geoffrey J. Gordon
Manfred K. Warmuth	Gerald Tesauro	Gerald Tesauro

Table 5. The number of publications by the relatively senior researcher and the relatively junior researcher and the number of their co-authored publications for the topic 'database systems' at Microsoft.

Senior	Junior	Co-authored
47	23	11
64	16	9
62	18	11
94	28	22

the clustering-based additive model is able to adjust the weights of the two types researchers. Similar clusters are also observed for the topic 'Machine Learning'.

Due to lack of ground-truth, we do not compare the performance of our clustering algorithm with other clustering algorithm such as modularity based clustering, K-means etc. However, our experimental results clearly show that our method is able to capture the senior-junior groups accurately.

4.3 Influence of Universities to Company

We next show our experiments on the influence of universities to companies. We show the top-5 most influential universities to the company 'Microsoft' on Data Mining and to the company 'IBM' on Database Systems in Table 6. It is fairly hard to determine if a rank is good or not since different people have different criteria and therefore there is even no ground-truth for comparison. What's more, our data set is not complete and lots of affiliation information for researchers is missing. Therefore, we do not report a detailed analysis of our ranking in this work. By simply looking at the ranks, we can see the universities that are well-known for their research in data mining and database systems such as 'wisc', 'uiuc', 'cmu', 'berkeley' etc. are ranked high in their influence to the two companies. Some other universities such as 'uic' also have high ranks. The National Center for Data Mining of University of Illinois Chicago is the member of the Data Mining Group [1] and therefore they have good connections with industry

Table 6. The top-5 most influential schools/research institutions to the company Microsoft on data mining and to the company IBM on database systems under different models. The schools are ranked by their influences under different models.

Data Mining (Microsoft)			Database Systems(IBM)		
simple additive	*weighted additive*	*clustering-based additive*	*simple additive*	*weighted additive*	*clustering-based additive*
uiuc	asu	ucr	berkeley	wisc	wisc
cmu	ucr	cmu	wisc	berkeley	berkeley
uic	cmu	asu	uiuc	toronto	toronto
asu	uic	uic	toronto	umd	umd
ucsb	uiuc	uci	umd	uiuc	uiuc

companies. Again, the rank is completely based on our current dataset and may not be accurate due to the missed affiliation information.

4.4 Simulated Data

Due to lack of ground-truth and missing affiliation information, we are not able to validate and compare the three models we proposed on the real data set. Thus we conduct the following set of experiments on simulated data. For simplicity, we only consider one topic. In our simulation, we generate a company with 200 researchers, which is comparable to the number of researchers of the real companies. We randomly select 20 of them as managers who are relatively influential in the company. For each manager, we assign the remaining researchers to his/her group randomly and we set up a *significant influence threshold* and a *low influence threshold* as 0.5 and 0.2, respectively. We assume the managers have significant influence to their group members thus we randomly generate influence from the managers to the other group members. The influences are in the range of [0.5, 1]. Each manager and his/her group then naturally represent a cluster.

Then we generate 400 researchers in the academia. Our ground-truth is we have two types of researchers in academia that have high influence to the company: (1) the researchers in academia who influence many researchers of the company (we call this type of researchers *influence many* researchers). (2) the researchers in academia who influence "important" researchers, namely managers of the company (we call this type of researchers *influence important* researchers). For type one researchers, we set up an *influence many threshold* as 30, where these researchers influence at least 30 researchers of the company. The influences, however, are all below the low influence threshold 0.2. For type two researchers, we set up an *influence important threshold* as 3, where these researchers influence at least 3 managers. The influences are above the significant influence threshold 0.5 and below 1.

We also generate another group of 20 academia researchers who influence researchers in the same group of the company. We call this set of researcher *influence same group* researchers. The motivation is they neither influence many researchers of the company nor any manager of the company. Therefore they do not belong to the influential researchers to the company. However, they do have significant influence to certain amount of researchers in the same group of the company. This is exactly the same situation as shown in Figure 1. As we discussed before, the simple additive model and the weighted additive model may not be able to distinguish them from the real influential researchers

since both models do not consider the relationship of the researchers of the company. On the contrary, the clustering-based additive model may be able to tell the influence is only on a small group of researchers who collaborate quite often, rather than company-wise influence. To validate this, we test the case where the group of researchers collaborating quite often, namely they have significant influence to each other as well as the case the group of researchers collaborating less often, namely they have low influence to each other.

The reason that we choose the above parameter settings is that these parameters are able to illustrate the effectiveness of the clustering-based additive model. With these parameters, both influence many and influence important researchers are indeed very influential, while the influence same group researchers may or may not be influential, depending on the correlation of the researchers in the group. With too extreme parameter settings, such as significant influence threshold as 0.9, or influence important threshold as 10, there might be no chance for the influence same group researchers to be as influential as the two types of researchers who are truly influential.

To compare the performance of difference models, we rank the researchers according to their influence to the company by the three different models we proposed. We then evaluate whether the ranks of the influential researchers to the company are indeed high or not. For the 10 "influence many" researchers and the 10 "influence important" researchers, we expect them to be ranked as the top-20 most influential researchers. Therefore if their ranks drop below 20, we think the model makes errors. As to the 20 "influence same group" researchers, we expect them to be ranked below the "influence important" and the "influence many" researchers. Therefore if their ranks are above 20, we think the model makes errors. We randomly simulate 10 data sets and show the averaged number of errors by each model for each type of researchers. The results are shown in Figure 5.

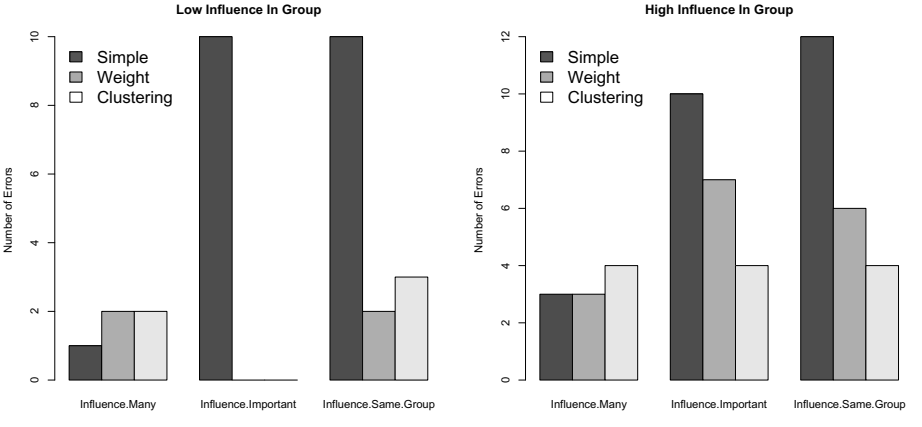

Fig. 5. Comparison of the performance of the three models

As we can see, when the group of researchers have low influence to each other, the ranks of the "influence many" researchers are all good for all three models. The ranks of the "influence important" researchers are good for the weighted additive model and the clustering-based additive model. The ranks are bad for the simple additive model since the "influence important" researchers only influence three researchers in the company, and the model doesn't consider the importance of these researchers. The ranks of the "influence same group" researchers are bad for the simple additive model as expected since they do not influence many researchers. The ranks for the weighted additive model and the clustering based additive model are comparable and are both good.

When the group of researchers have significant influence to each other, the performance of all three models drop for all three types of researchers since it becomes harder to distinguish the "influence same group" researchers with the "influence many" and the "influence important" researchers. For the "influence many" researchers, the ranks of all three models are still ok. For the "influence important" researchers, the ranks of the simple and weighted additive models are both bad since both models think the "influence same group" researchers are really influential to the company. The clustering-based additive model, on the contrary, integrates the information that the researchers of the company being influenced are from the same group and collaborate with each other quite often, and thus obtain relatively better ranks. For the same reason, the clustering-based additive model again achieves the best performance for the "influence same group" researchers.

As a conclusion, the simple additive model tends to assign high ranks to the "influence many" researchers, or the researchers who influence many researchers of the company and the influences are not necessarily significant. The weighted additive model tends to assign high ranks to the "influence important" researchers, or the researchers who influence only a few but important researchers of the company. The clustering-based additive model is not very different from the weighted additive model if the researchers of the company within the same group do not have significant influence to each other. However, when these researchers do have significant influence to each other, the clustering-based additive model has higher accuracy to assign relatively high ranks to the "influence important" researchers and relatively low ranks to the "influence same group" researchers.

5 Conclusion

In this work, we addressed the problem of mining research topic-specific influence between academia and industry. Based on the influence between individual researchers, we proposed three models – simple additive model, weighted additive model, clustering-based additive model – to learn the influence of individual researcher in academia to a company on specific research topics. We further derived the topic-specific influence from a research institution to a company. The influence from industry to academia can be obtained using the same models but with reverse influence direction. In our future work, we'd like to manually complete the missing affiliation information of researchers such that our experimental results are more accurate and the ranks we obtained are more meaningful.

Acknowledgements. The author wants to thank Dr. Tang for sharing the data set used in the work of [13].

References

1. Data Mining Group (2011), `http://www.dmg.org/`
2. Akaike, H.: A new look at the statistical model identification. IEEE Transactions on Automatic Control 19(6), 716–723 (2002)
3. Brandes, U., Delling, D., Gaertler, M., et al.: On modularity clustering. IEEE Transactions on Knowledge and Data Engineering, 172–188 (2007)
4. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 199–208. ACM, New York (2009)
5. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 57–66. ACM, New York (2001)
6. Goyal, A., Bonchi, F., Lakshmanan, L.V.S.: Learning influence probabilities in social networks. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 241–250. ACM, New York (2010)
7. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146. ACM, New York (2003)
8. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Transactions on Information Theory 47(2), 498–519 (2001)
9. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p. 429. ACM, New York (2007)
10. Rodriguez, M.G., Leskovec, J., Krause, A.: Inferring networks of diffusion and influence. In: KDD 2010 (2010)
11. Saito, K., Nakano, R., Kimura, M.: Prediction of information diffusion probabilities for independent cascade model. In: Knowledge-Based Intelligent Information and Engineering Systems, pp. 67–75. Springer, Heidelberg (2010)
12. Steyvers, M., Griffiths, T.: Probabilistic topic models. In: Handbook of latent semantic analysis, p. 427 (2007)
13. Tang, J., Sun, J., Wang, C., Yang, Z.: Social influence analysis in large-scale networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 807–816. ACM, New York (2009)
14. Wu, X., Zhang, C., Zhang, S.: Database classification for multi-database mining. Information Systems 30(1), 71–88 (2005)