Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2006-050 July 4, 2006

Iterative Collaborative Ranking of Customers and Providers

Loo Nin Teow, Dina Katabi

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Iterative Collaborative Ranking of Customers and Providers

Loo Nin Teow
DSO National Laboratories
20 Science Park Drive Singapore 118230
tloonin@dso.org.sg

Dina Katabi
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge MA 02139
dina@csail.mit.edu

Abstract

This paper introduces a new application: predicting the Internet provider-customer market. We cast the problem in the collaborative filtering framework, where we use current and past customer-provider relationships to compute for each Internet customer a ranking of potential future service providers. Furthermore, for each Internet service provider (ISP), we rank potential future customers. We develop a novel iterative ranking algorithm that draws inspiration from several sources, including collaborative filtering, webpage ranking, and kernel methods. Further analysis of our algorithm shows that it can be formulated in terms of an affine eigenvalue problem. Experiments on the actual Internet customer-provider data show promising results.

1 Introduction

In the competitive world of business, market targeting is an essential strategy. A provider of products or services would find it useful to know who will most likely become a customer, and which potential customers may be drawn towards one's competitors. Conversely, that provider might be interested to know how customers rate the various providers. Both of these can be cast as ranking problems, i.e., ranking of potential customers for a particular provider, or ranking of potential providers for a particular customer.

Typically, the actual ranking of customers and providers is based on business considerations such as costs, quality of product or service, geographic location, etc., together with market models that depend on the type of business. This information however is usually not open to the public and differs for each business and customer-provider pair.

Nonetheless, one can still perform ranking based on choices already made. These include the customers that a provider already has, or the providers that a customer already chose. A key assumption is that if other customers similar to oneself (in terms of common providers) have chosen a particular provider, then one is likely to choose that provider as well. This is very similar to the idea underlying collaborative filtering [1][2], with users and items being analogous to customers and providers respectively.

As we shall show, collaborative filtering alone gives sub-optimal results on the above problem because it does not fully exploit the topology of the network of customers and providers. To do this, we have developed an iterative algorithm in which collaborative filtering is but the first iteration. Our algorithm is inspired by the HITS webpage ranking algorithm [3], with hubs and authorities being analogous to customers and providers respectively. The algorithm captures similarities between customers (or providers) using kernel functions [4], and recursively updates their provider (or customer) preferences until they converge to a fixed assignment. Our algorithm can be formulated in terms of an affine eigenvalue problem, thereby allowing us to infer its convergence properties.

The contributions of this paper are twofold. First, the paper introduces the novel application of predicting the Internet provider-customer market. It casts the problem within the collaborative filtering framework, and uses actual provider-customer data extracted from the Internet routing system to show that the current Internet graph embodies valuable information about future provider-customer relationships. Second, the paper introduces a novel iterative ranking algorithm that extends prior user-based and item-based collaborative filtering algorithms to capture the relation between items and users in a recursive manner. When applied to the actual Internet provider-customer data, the new algorithm produces an accuracy of 83% for ranking customers, whereas a user-based collaborative filtering approach produces an accuracy of 71%. As for ranking providers, the new algorithm produces an accuracy of 91%, whereas an item-based collaborative filtering approach produces an accuracy of 84%. Finally, the developed ranking algorithm is generic and can be applied to other collaborative filtering problems.

2 Related Fields

The algorithm that we have developed is a synergy of key ideas from several fields: collaborative filtering [1][2], webpage ranking [3][5], and kernel methods [4].

2.1 Collaborative Filtering

Collaborative filtering is an automated method for predicting user preferences for a set of items using the preference information of other users. The preference information may be implicitly encoded in past choices. Collaborating filtering is often used in recommendation systems, which offer suggestions such as "users who bought item *i* also bought item *j*".

There are two main types of collaborative filtering: user-based and item-based [2]. User-based algorithms represent a user as a vector in the item-space and compute similarities between users given the items they chose, while item-based algorithms represent an item as a vector in the user-space and identify the similarity between two items by comparing users' choices on them. Then, given user *u* and item *i*, one can compute the preference as follows:

User-based: $$preference(u,i) = \frac{\sum_{v}^{\text{users}} similarity(u,v) * chosen(v,i)}{\sum_{v}^{\text{users}} similarity(u,v)}$$

Item-based: $$preference(u,i) = \frac{\sum_{j}^{\text{items}} similarity(i,j) * chosen(u,j)}{\sum_{j}^{\text{items}} similarity(i,j)}$$

The similarity function is usually computed as the normalized dot product or Pearson correlation between the users' item vectors or the items' user vectors. These are binary vectors that encode past choices of items made by users. The function "chosen" returns 1 or 0 depending on whether the user has previously chosen the item. Collaborative filtering does not differentiate between ranking of users and ranking of items.

2.2 Webpage Ranking

Ranking webpages returned as a result of a query is an essential function of search engines. The two most frequently cited webpage ranking algorithms are HITS [3] and PageRank [5]. Since our algorithm is closer in principle to HITS, we shall briefly describe this method.

HITS (Hyperlink Induced Topic Search) defines two kinds of web pages: authorities and hubs. An authority is a webpage with several in-links, while a hub has several out-links. The main idea underlying HITS is that good hubs point to good authorities, and this is necessarily recursive since a webpage can be both a hub and an authority. Without going through the derivation, this idea may be realized as the following iterative algorithm.

Given n web pages, let $\mathbf{L}$ be the n x n adjacency matrix of the directed web graph, where each node in the graph represents a webpage, and

$$\mathbf{L}_{ij} = \begin{cases} 1 & \text{if there is a link from node } i \text{ to node } j \\ 0 & \text{otherwise} \end{cases}$$

HITS assigns a hub score and an authority score to each webpage iteratively as follows:

$$\mathbf{x}(t) = \mathbf{L}^T \mathbf{L}\,\mathbf{x}(t-1) \qquad \mathbf{x}(t) \leftarrow \frac{\mathbf{x}(t)}{\|\mathbf{x}(t)\|}$$

$$\mathbf{y}(t) = \mathbf{L}\mathbf{L}^T\,\mathbf{y}(t-1) \qquad \mathbf{y}(t) \leftarrow \frac{\mathbf{y}(t)}{\|\mathbf{y}(t)\|}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the vectors of authority scores and hub scores respectively. $\mathbf{L}^T\mathbf{L}$ and $\mathbf{L}\mathbf{L}^T$ are known as the authority matrix and hub matrix respectively. This algorithm is proven to converge to fixed values as $t \to \infty$.

2.3 Kernel Methods

Kernel methods [4] such as Support Vector Machines and Gaussian Processes have recently been popular in the machine learning literature. This is due in part to their ability to implicitly map input data to high dimensional feature space through kernel functions. Kernel functions also allow us to define similarities between objects of any kind, while satisfying certain properties that allow further analysis to be performed in feature space.

If we restrict ourselves to input data in vector form, then letting $\mathbf{x} \in \Re^p$ be an input vector, we have a mapping $\phi : \Re^p \to \Re^q$, where $\Re^q$ is a high-dimensional feature space. In kernel methods, we do not have to define ϕ explicitly. Instead, we employ what is known as the "kernel trick", by expressing the dot product in feature space as a kernel function:

$$\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j)$$

Examples of widely used kernel functions are:

Gaussian: $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma\, |\, \mathbf{x}_i - \mathbf{x}_j\, |^2\right)$, Polynomial: $k(\mathbf{x}_i, \mathbf{x}_j) = \left(\mathbf{x}_i \cdot \mathbf{x}_j\right)^d$,

where $\gamma \in \Re$ and $d \in Z^+$ are the kernel parameters.

3 Application Domain: Predicting the Evolution of Internet Customer-Provider Relationships

The Internet is composed of multiple networks, usually referred to as autonomous systems (ASs). An autonomous system (AS) is effectively a set of hosts and routers under a single administration [6]. There are several types of ASs as well as several types of relationships between ASs. It is common to divide ASs into Internet Service Providers (ISPs), (e.g., AT&T, Quest, and Level3), which provide transit Internet service to other networks, and stub-networks (e.g., MIT, Berkeley, and Harvard), which carry only their own traffic. Furthermore, depending on their size and geographical span, ISPs themselves are divided into international, national, regional and local.

The business relationships between the ASs define how both traffic and money flow in the Internet. A pair of ASs may be connected using a customer-provider relationship, where the provider agrees to carry traffic from and to the customer network, in return for a monthly payment. Alternatively, a pair of ASs may have a peering relationship, in which both ASs agree to carry traffic for each other without any exchange of money.

In this paper, we are primarily interested in customer-provider relationships, which constitute about 80%-90% of inter-AS relationships. Thus, we consider the ***AS-graph*** where each node is an AS, and each directed edge refers to a customer-provider relationship. A provider may have a number of customers that varies between 1 to a few thousands. An Internet customer may have multiple providers. Small providers are themselves customers of bigger providers. Additionally, the AS-graph changes over time as new customers and providers are created or old ASs change their business relationships.

This paper addresses the following problem: given the recent history of the AS-graph, for each customer, predict a ranking of all providers that correlates well with the customer's future providers. We are also interested in the dual problem which can be stated as follows: for each provider, compute a ranking of all customers that correlates well with the provider's soon-to-be customers.

4 Iterative Collaborative Ranking

Given n customer and m providers, let $\mathbf{A}$ be the n x m adjacency matrix (of the AS-graph), such that:

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if customer } i \text{ is linked to provider } j \\ 0 & \text{otherwise} \end{cases}$$

Let $\mathbf{cust}_i$ and $\mathbf{prov}_j$ be the customer and provider vectors corresponding to the i^{th} row and j^{th} column of matrix $\mathbf{A}$ respectively. Further let $\mathbf{G}^{(\text{cust})}$ be the customer-based n x n Gram matrix, and $\mathbf{G}^{(\text{prov})}$ be the provider-based m x m Gram matrix, such that:

$$\mathbf{G}^{(\text{cust})}_{ab} = k(\mathbf{cust}_a, \mathbf{cust}_b)\,, \qquad \mathbf{G}^{(\text{prov})}_{ab} = k(\mathbf{prov}_a, \mathbf{prov}_b)\,,$$

where k is a kernel function measuring the similarity between customers (providers) in terms of their common providers (customers). We have tried various types of kernel functions such as the normalized dot product and the Gaussian kernel. So far, we have found the best performing kernel function to be the polynomial kernel:

$$k(\mathbf{cust}_a, \mathbf{cust}_b) = (\mathbf{cust}_a \cdot \mathbf{cust}_b)^d\,, \qquad k(\mathbf{prov}_a, \mathbf{prov}_b) = (\mathbf{prov}_a \cdot \mathbf{prov}_b)^d\,,$$

where d is a positive integer. This is simply the number of common customers or providers raised to the d^{th} power.

Subtracting the diagonal from each gram matrix and then normalizing each row to sum up to one, we have the following stochastic matrices:

$$\mathbf{H}^{(\text{cust})}_{ab} = \begin{cases} 0 & \text{if } a = b \\ \dfrac{\mathbf{G}^{(\text{cust})}_{ab}}{\sum_{i=1, i\neq a}^{n} \mathbf{G}^{(\text{cust})}_{ai}} & \text{if } a \neq b \end{cases} \quad ; \qquad \mathbf{H}^{(\text{prov})}_{ab} = \begin{cases} 0 & \text{if } a = b \\ \dfrac{\mathbf{G}^{(\text{prov})}_{ab}}{\sum_{j=1, j\neq a}^{m} \mathbf{G}^{(\text{prov})}_{aj}} & \text{if } a \neq b \end{cases}$$

Note that each row in $\mathbf{H}^{(\text{cust})}$ refers to a particular customer; it presents the customer's fractional similarity to other customers in the system. The entries in each row sum up to 1, indicating that each customer has a total of one unit of similarity, divided among other customers in the system ($\mathbf{H}^{(\text{prov})}$ is the analogous counterpart for the providers.)

Now, we can rank all customers from the perspective of provider J as $\mathbf{H}^{(\text{cust})}\mathbf{prov}_J$, where $\mathbf{prov}_J$ is the J^{th} column in the adjacency matrix $\mathbf{A}$. Note that this is similar to the simple user-based collaborative filtering. Analogously, $\mathbf{H}^{(\text{prov})}\mathbf{cust}_I$ ranks all providers from the perspective of customer I and is similar to item-based collaborative filtering.

But our algorithm does not stop here; inspired by the HITS approach, it iterates on these rankings until they converge. However, the crucial difference from HITS is that as we iterate we "clamp" the values corresponding to existing customers or providers, and perform the vector normalization only for those values that do not correspond to current customers (providers). The intuition is that we want to compute a ranking that reflects the probability of connecting to future providers (customers), and we would like to iterate on these probabilities until they converge. However, as we iterate we would like to ensure that the probabilities of connecting to existing providers (customers) are kept at 1. This can be done by applying a diagonal masking matrix $[\mathbf{I}-\text{diag}(\mathbf{prov}_J)]$ or $[\mathbf{I}-\text{diag}(\mathbf{cust}_I)]$ before the vector normalization, and then adding back $\mathbf{prov}_j$ or $\mathbf{cust}_i$ after normalizing.

Specifically, to rank potential customers for provider *J*:

Let $\mathbf{v}(0)=\mathbf{prov}_J$ be the initial vector of customer scores.

Iterate for $t=1,\ldots$, until $\mathbf{v}$ converges:

$$\mathbf{u}(t)=[\mathbf{I}-\text{diag}(\mathbf{prov}_J)]\mathbf{H}^{(\text{cust})}\mathbf{v}(t-1)$$

$$\mathbf{v}(t)=\frac{\mathbf{u}(t)}{\|\mathbf{u}(t)\|_p}+\mathbf{prov}_J$$

Likewise, to rank potential providers for customer *I*:

Let $\mathbf{v}(0)=\mathbf{cust}_I$ be the initial vector of provider scores.

Iterate for $t=1,\ldots$, until $\mathbf{v}$ converges:

$$\mathbf{u}(t)=[\mathbf{I}-\text{diag}(\mathbf{cust}_I)]\mathbf{H}^{(\text{prov})}\mathbf{v}(t-1)$$

$$\mathbf{v}(t)=\frac{\mathbf{u}(t)}{\|\mathbf{u}(t)\|_p}+\mathbf{cust}_I$$

In either case, parameters to set are *d* for the kernel function and *p* for the vector norm.

Intuitively, each iteration's ranking scores serves to reinforce those in the next iteration until they stabilize. Like HITS, this is somewhat similar to the power method for solving the eigenvalue problem $\lambda\mathbf{v}=\mathbf{H}\mathbf{v}$, except that values corresponding to existing customers or providers are "clamped" at ones throughout. Note that standard collaborative filtering is equivalent to $\mathbf{v}(t = 1)$ (i.e., it stops at the first iteration).

4.1 Affine Eigenvalue Problem

We show that our algorithm solves an affine eigenvalue problem and thus converges to fixed values. Let $\mathbf{y}$ be either $\mathbf{cust}_I$ or $\mathbf{prov}_J$, a vector containing l zeroes, and let $\mathbf{z} = \mathbf{H}\mathbf{y}$. Construct a l x l matrix $\mathbf{M}$ by excluding those rows and columns in $\mathbf{H}$ corresponding to the ones in $\mathbf{y}$. Similarly construct a l x 1 vector $\mathbf{c}$ by excluding those entries in $\mathbf{z}$ corresponding to the ones in $\mathbf{y}$. Then our iterative algorithm can be reformulated as follows:

$$\mathbf{x}(0)=\mathbf{0}$$

$$\mathbf{x}(t)=\frac{\mathbf{M}\mathbf{x}(t-1)+\mathbf{c}}{\|\mathbf{M}\mathbf{x}(t-1)+\mathbf{c}\|_p} \quad \text{for } t=1,\ldots$$

This is equivalent to solving what is known as an affine eigenvalue problem: $\lambda\mathbf{x}=\mathbf{M}\mathbf{x}+\mathbf{c}$.

In [7], it is proven that if $\mathbf{M}$ and $\mathbf{c}$ are both nonnegative, such that $\mathbf{M}\mathbf{x}+\mathbf{c}>\mathbf{0}$ for all $\mathbf{x} > \mathbf{0}$, then for a monotone vector norm, the above iterations will converge to give a unique solution regardless of the initial vector $\mathbf{x}(0)$. The first condition is true in our case by virtue of the nonnegative kernel function, while the *p*-norm is a monotone vector norm. It can be shown that by applying Google's "trick" in PageRank [5] to the collaborative matrix $\mathbf{H}$:

$$\mathbf{H} \leftarrow \alpha \mathbf{H} + (1-\alpha)\tfrac{1}{\beta}\mathbf{1}\mathbf{1}^T \qquad \text{where } 0 < \alpha < 1, \beta \text{ is the number of rows in } \mathbf{H},$$

and $\mathbf{1}$ is the vector of ones, then the final condition, $\mathbf{Mx}+\mathbf{c}>\mathbf{0}$ for all $\mathbf{x} > \mathbf{0}$, is enforced, thereby guaranteeing convergence to a unique solution regardless of the initial vector $\mathbf{x}(0)$. In our experiments, we found a good value of α to be 0.99.

5 Autonomous System Dataset

At each AS, a Border Gateway Protocol (BGP) [8] routing table is used to coordinate routing of data through the network. BGP routing tables are publicly available from Oregon Route Views at http://www.routeviews.org/ and RIPE at http://ww.ripe.net/. We can infer AS customer-provider relationships from the BGP data, by using an algorithm called PTE (Partialness To Entireness) [9]. The software for this algorithm can be downloaded from http://rio.ecs.umass.edu/download.htm. The inferred relationships are then used to construct daily (directed) customer-provider networks, in which two ASs are linked if one is the customer of the other on that day.

In order to have more accurate views of the AS-graph, we have combined multiple BGP routing tables from Oregon Route Views (Views2) and RIPE (rrc00). Specifically, we have constructed AS-graphs for all days in the year 2003, with each day having 12 BGP tables from Oregon Route Views (Views2) and 3 BGP tables from RIPE (rrc00). Oregon Route Views has missing data on 1 June 2003, and hence we exclude this day. The reason for choosing the year 2003 is because PTE exploits known AS relationships, and the only ones available were manually collected from various sources in 2003.

We have found that an appropriate time scale for our problem is on a monthly basis. As such, we have collapsed all the daily AS-graphs in each month into a single graph, while removing contradictory links. Those links removed include bi-directional links (since customer-provider relationships are by definition uni-directional), as well as links that have corresponding peering edges (since inter-AS relationships cannot be both customer-provider and peering at the same time).

The final result is a set of 12 AS-graphs, one for each month of 2003. A monthly AS-graph would typically have around 16,000 customers and 2,500 providers, giving a total of about 40 million AS pairs, of which only about 30,000 (0.075%) are linked. Hence, the corresponding adjacency matrix is extremely large but also extremely sparse.

6 Experiments

In this section, we evaluate the predictive power of our iterative ranking algorithm and compare it with traditional user-based and item-based collaborative filtering. We arbitrarily select the AS-graphs of Sep 2003 and Oct 2003, for our experiments. We compute the rankings based on Sep 2003, and then see if these rankings correlate with new customer-provider relationships created in the next month, Oct 2003.

A good ranking should rank the actual customer-provider relationships created in Oct 2003 higher than those that did not occur in Oct 2003. For example, say that in Sep 2003, customer I is neither connected to Quest nor AT&T, but in Oct 2003, it buys service from Quest. A good ranking algorithm operating on the AS-graph in Sep 2003 should rank Quest higher than AT&T from the perspective of customer I. Thus, we evaluate the accuracy of a ranking algorithm using the following performance metric:

$$Q^{(\text{cust})} = \frac{1}{m'}\sum_{j=1}^{m'} \frac{1}{N_j(n_j - N_j)} \sum_{a\in\{\text{new}\}}^{N_j} \sum_{b\in\{\text{not new}\}}^{n_j - N_j} \delta(rank_{ja} > rank_{jb})$$

$$Q^{(\text{prov})} = \frac{1}{n'}\sum_{i=1}^{n'} \frac{1}{M_i(m_i - M_i)} \sum_{a\in\{\text{new}\}}^{M_i} \sum_{b\in\{\text{not new}\}}^{m_i - M_i} \delta(rank_{ia} > rank_{ib})$$

Here, n' is the number of customers that have new providers in Oct 2003, m' is the number of providers that have new customers in Oct 2003, n_j and N_j are the numbers of potential and new customers respectively in Oct 2003 for provider j, m_i and M_i are the numbers of potential and new providers respectively in Oct 2003 for customer i, and $\delta()$ is a indicative function which is set to 1 if the statement in brackets is true and is 0 otherwise. The above metric computes the proportion of "new, not-new" pairs that are correctly ranked, averaged over the providers (or customers). It attains the maximum of 1 when all new customers (providers) acquired in Oct 2003 are ranked above all non-customers (non-providers), and the minimum of 0 when all new customers (providers) acquired in Oct 2003 are ranked below all non-customers (non-providers). If the ranking is perfectly random and uncorrelated to the customer-provider relationships created in Oct 2003, the metric is 0.5.

To evaluate the effectiveness of our algorithm, we consider the new customer-provider relationships created in Oct 2003. 482 providers (out of the 2567 that existed in Sep 2003) acquired new customers in Oct 2003, resulting in a total of 1518 new customers. Also, 1379 customers (out of the 16114 that existed in Sep 2003) added a new provider in Oct 2003.

We try the following parameter values for the degree of the polynomial kernel and the normalization norm: $d = 1,\ldots,10$, and $p = 1, 2, 10, 20$, and ∞. From our experiments, we find that for ranking customers, $d = 8$ and $p = 2$ give the best performance of 83%, whereas for ranking providers, $d = 2$ and $p = 1$ give the best performance of 91%. Higher degrees probably result in overfitting, while the converse is true for lower degrees. Figures 1 and 2 show the results of ranking customers for fixed $d = 8$ (varying p) and fixed $p = 2$ (varying d) respectively, while Figure 3 and 4 show the results of ranking providers for fixed $d = 2$ (varying p) and fixed $p = 1$ (varying d) respectively. For clarity, we show only the graphs for even numbers of d.

We make the following observations:

- The performance at $t = 1$ is seldom if ever the best one. Further iterations typically bring about an improvement in performance. This means that the iterative algorithm outperforms traditional user-based and item-based collaborative filtering on the data.
- Every degree d has an optimal p-norm. Infinity-norm tends to perform badly compared with other p-norms, at least for the values of d that we tried. In addition, each provider or customer has its own optimum combination of d and p.
- Convergence is relatively fast, especially for small values of d and p; five or six iterations are usually sufficient.
- Finally, our results show that the current AS-graph embodies valuable information about future provider-customer business relationships. Since the AS-graph is publicly available, ISPs might find this information useful in targeting potential customers and predicting the future business relationships of their competitors.

7 Further Discussion and Conclusion

We have combined key ingredients from collaborative filtering, webpage ranking, and kernel methods into a novel algorithm for ranking Internet customers and providers. Furthermore, we have shown that our algorithm gives promising results for predicting future customer-provider business relationships.

Our work can be extended in multiple directions. First, it would be interesting to study the performance of the proposed iterative algorithm on more traditional collaborative filtering problems and datasets, and compare it against a larger set of collaborative filtering algorithms beyond the user-based and item-based approaches. Since kernel functions can be combined to form new kernel functions, the new algorithm may allow us to incorporate various factors that are relevant to the domain in a strategy similar to that used in [10].

Second, it would be interesting to examine the implications of our algorithm to the webpage ranking problem. Since the new algorithm relies on kernel functions, it allows us to replace the authority and hub matrices in HITS with Gram matrices, in which case HITS would become equivalent to finding the dominant eigenvectors in kernel space.

Third, our algorithm can be adapted to other kinds of networks such as social networks, where we can use the algorithm for ranking potential friends. Since a social network is undirected, its adjacency matrix is symmetric, and hence there will be just one set of rankings rather than two. The ranking scores generated by our algorithm should also serve as good features for the prediction of future friendships.

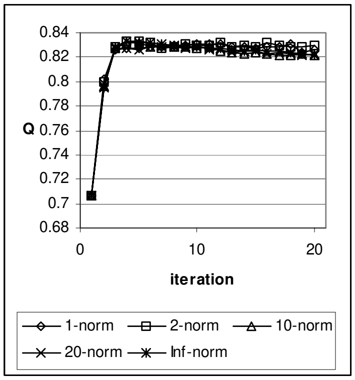

Figure 1: Ranking customers, degree = 8

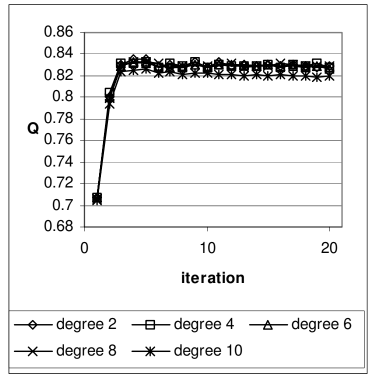

Figure 2: Ranking customers, 2-norm.

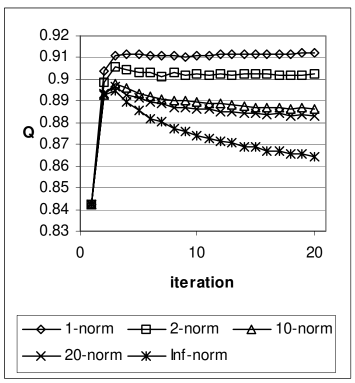

Figure 3: Ranking providers, degree = 2.

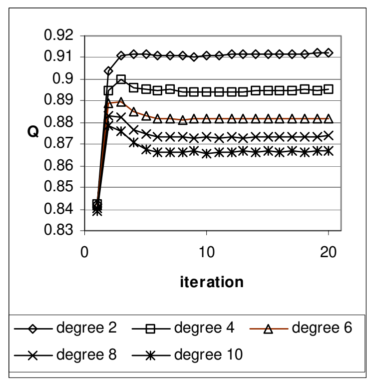

Figure 4: Ranking providers, 1-norm.

References

[1] Goldberg, D., Nichols, D., Oki, B.M., & Terry, D. (1992) Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35:12, pp.61-70.

[2] Sarwar, B.M., Karypis, G., Konstan, J.a., & Reidl, J. (2001) Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th International World Wide Web Conference*, pp.285-295.

[3] Kleinberg, J. (1999) Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46, pp.604-632.

[4] Scholköpf, B., & Smola, A.J. (2001) *Learning with kernels*. Cambridge, MA: MIT Press.

[5] Brin, S., Page, L., Motwami, R., & Winograd, T. (1999) The PageRank Citation Ranking: Bringing Order to the Web, *Technical Report 1999-0120*, Computer Science Department, Stanford University, Stanford, CA.

[6] Halabi, S., & McPherson, D. (2001) *Internet Routing Architectures*, 2nd ed, Cisco Press.

[7] Blondel, V.D., Ninove, L., & Dooren, P.V. (2005) An affine eigenvalue problem on the negative orthant. *Linear Algebra and its Applications*, 404, pp.69-84.

[8] Stewart, J.W. (1998) *BGP4: Inter-Domain Routing in the Internet*, Addison-Wesley.

[9] Xia, J., & Gao, L., (2004) On the Evaluation of AS Relationship Inferences. *Proceedings of IEEE Global Communications Conference (GLOBECOM'04)*, 3, pp1373-1377.

[10] Basilico, J., & Hofmann, T. (2004) Unifying Collaborative and Content-Based Filtering. *Proceedings of the 21st International Conference on Machine Learning*, pp.65-72.