The work of Jon Kleinberg

John Hopcroft

Introduction

Jon Kleinberg's research has helped lay the theoretical foundations for the information age. He has developed the theory that underlies search engines, collaborative filtering, organizing and extracting information from sources such as the World Wide Web, news streams, and the large data collections that are becoming available in astronomy, bioinformatics and many other areas. The following is a brief overview of five of his major contributions.

Hubs and authorities

In the 1960s library science developed the vector space model for representing documents [13]. The vector space model is constructed by sorting all words in the vocabulary of some corpus of documents and forming a vector space model where each dimension corresponds to one word of the vocabulary. A document is represented as a vector where the value of each coordinate is the number of times the word associated with that dimension appears in the document. Two documents are likely to be on a related topic if the angle between their vector representations is small. Early search engines relied on the vector space model to find web pages that were close to a query. Jon's work on hubs and authorities recognized that the link structure of the web provided additional information to aid in tasks such as search. His work on hubs and authorities addressed the problem of how, out of the millions of documents on the World Wide Web (WWW), you can select a small number in response to a query. Prior to 1997 search engines selected documents based on the vector space model or a variant of it. Jon's work on hubs and authorities [5] laid the foundation to rank pages based on links as opposed to word content. He introduced the notion of an authority as an important page on a topic and a hub as a page that has links to many authorities. Mathematically, an authority is a page pointed to by many hubs and a hub is a page that points to many authorities. For the concepts of hubs and authorities to be useful, one needs to develop the mathematics to identify hubs and authorities; that is, to break the cycle in the definition.

The World Wide Web can be represented as a directed graph where nodes correspond to web pages and directed edges represent links from one page to another. Let A be the adjacency matrix for the underlying web graph. Jon did a text based search

Proceedings of the International Congress
of Mathematicians, Madrid, Spain, 2006

to find, say, the 200 most relevant web pages for a query based on word content. This set might not contain the most important web sites since, as he points out, the words "search engine" did not appear on the sites of the popular search engines in 1997, such as Alta Vista or Excite. Similarly there were over a million web sites containing the word "Harvard" but the site www.harvard.edu was not the site that contained the term "Harvard" most often. Thus, he expanded the set by adding web sites reached by either in or out links from the 200 sites. To avoid adding thousands of additional web sites when one of the web pages was extremely popular, he restricted each page in the set to add at most fifty additional pages to the original set. In the process of adding web pages, he ignored links to pages within the same domain name as these tended to be navigational links such as "top of page". In the resulting sub graph, which was now likely to contain most of the important relevant pages, he assigned weights to pages and then iteratively adjusted the weights. Actually, each page was assigned two weights, a hub weight and an authority weight. The hub weights were updated by replacing the weight of each hub with the sum of the weights of the authorities that it points to. Next the weights of the authorities were updated by replacing the weight of each authority with the sum of the hub weights pointing to it. The hub weights and the authority weights were then normalized so that the sum of the squares of each set of weights equaled one. This iterative technique converges so that the hub weights are the coordinates of the major eigenvector of AA^T and the authority weights are the coordinates of the major eigenvector of $A^T A$. Thus, the eigenvectors of AA^T and $A^T A$ rank the pages as hubs and authorities. This work allowed a global analysis of the full WWW link structure to be replaced by a much more local method of analysis on a small focused sub graph.

This work is closely related to the work of Brin and Page [2] that lead to Google. Brin and Page did a random walk on the underlying graph of the WWW and computed the stationary probability of the walk. Since the directed graph has some nodes with no out degree, they had to resolve the problem of losing probability when a walk reached a node with no out going edges. Actually, they had to solve the more general problem of the probability ending up on sub graphs with no out going edges, leaving the other nodes with zero probability. The way this was resolved was that at each step the walk would jump with some small probability ε to a node selected uniformly at random and with probability $1-\varepsilon$ take a step of the random walk to an adjacent node.

Kleinberg's research on hubs and authorities has influenced the way that all major search engines rank pages today. It has also spawned an industry creating ways to help organizations get their web pages to the top of lists produced by search engines to various queries. Today there is a broad field of research in universities based on this work.

Small worlds

We are all familiar with the notion of "six degrees of separation", the notion that any two people in the world are connected by a short string of acquaintances. Stanley Milgram [12] in the sixties carried out experiments in which a letter would be given to an individual in a state such as Nebraska with instructions to get the letter to an individual in Massachusetts by mailing it to a friend known on a first name basis. The friend would be given the same instructions. The length of the path from Nebraska to Massachusetts would typically be between five and six steps.

Milgram's experiments on this inter-personal connectivity lead to a substantial research effort in the social sciences focused on the interconnections in social networks. From 1967 to 1999 this work was primarily concerned with the structure of relationships and the existence of short paths in social networks. Although the fact that individuals have the ability to actually find the short paths as was demonstrated by Milgram's original experiment, there was no work on understanding how individuals actually found the short paths or what conditions were necessary for them to do so.

In 1998 Watts and Strogatz [14] refined the concept of a small world, giving precise definitions and simple models. Their work captured the intuitive notion of a reference frame, such as the geographical location where people live or their occupation. In this reference frame, an individual is more likely to know the neighbor next door than a person in a different state. Most people know their neighbors but they also know some people who are far removed. The relationships between individuals and their neighbors were referred to as short links, and the few friends or relatives far away that the individuals knew were referred to as long-range links.

One simple model developed by Watts and Strogatz was a circular ring of nodes where each node was connected to its nearest neighbors clockwise and counterclockwise around the circle, as well as to a few randomly selected nodes that were far away. Watts and Strogatz proved that any pair of nodes is, with high probability, connected by a short path, thus justifying the terminology "small world".

Jon [6] raised the issue of how you find these short paths in a social network without creating a map of the entire social world. That is, how do you find a path using only local information? He assumed a rectangular grid with nodes connected to their four nearest neighbors, along with one random long range connection from each node. As the distance increased the probability of a long range random link connecting two nodes decreased. Jon's model captured the concept of a reference frame with different scales of resolution: neighbor, same block, same city, or same country. Jon showed that when the decrease in probability was quadratic with distance, then there exists an efficient (polynomial time) algorithm for finding a short path. If the probability decreases slower or faster, he proved the surprising result that no efficient algorithm, using only local information, could exist for finding a short path even though a short path may exist.

In Jon's model the probability of a long range edge between nodes x and y decreased as $\mathrm{dist}(x, y)^{-r}$ where $\mathrm{dist}(x, y)$ is the grid distance between nodes x and y.

For $r = 0$, the probability of a long range contact is independent of distance. In this case, the average length of such a contact is fairly long, but the long range contacts are independent of the geometry of the grid and there is no effective way to use them in finding short paths even though short paths exist. As r increases, the average length of the random long range contacts decreases but their structure starts to become useful in finding short paths. At $r = 2$ these two phenomena are balanced and one can find short paths efficiently using only local knowledge. For $r > 2$ the average length of the random long-range contact continues to decrease. Although short paths may still exist, there is no polynomial time algorithm using only local information for finding them. When r equals infinity, no long-range contacts exist and hence no short paths. What is surprising is that for $r < 2$ or for $r > 2$, no efficient algorithm using only local information exists for finding short paths.

Theorem 1. *Let G be a random graph consisting of an $n \times n$ grid plus an additional edge from each vertex u to some random vertex v where the probability of the edge (u, v) is inversely proportional to $\mathrm{dist}(u, v)^r$. Here $\mathrm{dist}(u, v)$ is the grid distance between vertices u and v. For $r = 2$, there is a decentralized algorithm so that the expected time to find a path from some start vertex s to a destination vertex t is $O(\log^2 n)$.*

Proof. At each step the algorithm selects the edge from its current location that gets it closest to its destination. The algorithm is said to be in phase j when the lattice distance from the current vertex to the destination t is in the interval $(2^j, 2^{j+1})$. Thus, there are at most $\log n$ phases. We will now prove that the expected time the algorithm remains in each phase is at most $\log n$ steps and, hence, the time to find a path is $O(\log^2 n)$.

For a fixed vertex u the probability that the long-distance edge from u goes to v is

$$\frac{d(u, v)^{-2}}{\sum_{w \neq u} d(u, w)^{-2}}.$$

We wish to get an upper bound on the denominator so as to get a lower bound on the probability of an edge of distance $d(u, v)$. Since the set of vertices at distance i from u forms a diamond centered at u with sides of length i, there are $4i$ vertices at distance i from vertex u, unless u is close to a boundary in which case there are fewer. Thus

$$\sum_{w \neq u} d(u, w)^{-2} \leq \sum_{i=1}^{2n-2} 4i \frac{1}{i^2} = 4 \sum_{i=1}^{2n-2} \frac{1}{i}.$$

For large n there exists a constant c_1 such that

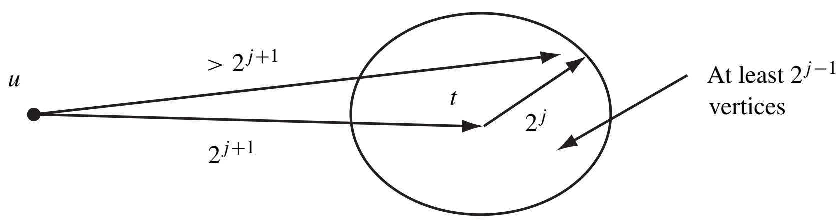
$$\sum_{w \neq u} d(u, w)^{-2} \leq c_1 \ln n.$$

It follows that there exists a constant c_2 such that each vertex that is within distance 2^j of u has probability of at least $c_2 \frac{2^{-2j}}{\ln n}$ of being the long distance contact of u.

The current step ends phase j of the algorithm if the vertex reached is within distance 2^j of the destination t. In the plane, the number of vertices at distance i from a given vertex grows linearly with i. Thus, the number of vertices within distance 2^j of the destination t is at least

$$\sum_{i=1}^{2^j} i = \frac{2^j(2^j+2)}{2} > 2^{2j-2}.$$

Since the current location is within distance 2^{j+1} of t and since there are at least 2^{2j-1} vertices within distance 2^j of t, there are at least 2^{2j-1} vertices within distance $2^{j+1}+2^j < 2^{j+2}$ of the current location. Each of these vertices that are within distance 2^{j+2} of the current location has probability of at least $\frac{c_2}{\ln(n)2^{2j+4}}$ of being the long-distance contact.



If one of the 2^{2j-1} vertices that are within distance 2^j of t and within distance 2^{j+2} of the current location is the long-distance contact of u, it will be u's closest neighbor to t. Thus, phase j ends with probability at least

$$\frac{c_2 2^{2j-1}}{\ln(n)2^{2j+4}} = \frac{c_2}{8\ln(n)}.$$

We now bound by $\log n$ the total time spent in step j. For $j \le \log\log n$ the current vertex is distance at most $\log n$ from the destination t. Thus, even taking only local edges suffices. For $j > \log\log n$, let x_j be the number of steps spent in phase j. Then

$$E(x_j) = \sum_{i=1}^{\infty} i \operatorname{Prob}(x_j = i).$$

Since

$$1 \operatorname{Prob}(x_i = 1) + 2 \operatorname{Prob}(x_i = 2) + \cdots = \operatorname{Prob}(x_i \ge 1) + \operatorname{Prob}(x_i \ge 2) + \cdots$$

we get

$$E(x_j) = \sum_{i=1}^{\infty} \operatorname{Prob}(x_j \ge i) \le \sum_{i=1}^{\infty} \left(1 - \frac{c_2}{8\ln(n)}\right)^{i-1} = \frac{1}{1-\left(1-\frac{c_2}{8\ln(n)}\right)} = \frac{8\ln n}{c_2}.$$

Thus, the total number of steps is $O(\log^2 n)$. □

Even more surprising than the above result that states there exists an efficient local algorithm for finding short paths when the exponent r equals two, were Jon's additional results proving no such algorithms exist when the exponent r was either greater than two or less than two.

This research on finding paths in small worlds has found applications outside the social sciences in such areas as peer-to-peer file sharing systems. It turns out that many real sets of data have the needed quadratic decrease in probability distribution. For example, an on-line community where you measure distance between individuals by the distance between their zip codes, often has this distribution after distances are corrected for the highly nonuniform population density of the U.S. [11].

Bursts

In order to understand a stream of information, one may organize it by topic, time, or some other parameter. In many data streams a topic suddenly appears with high frequency and then dies out. The burst of activity provides a structure that can be used to identify information in the data stream. Jon's work [7] on bursts developed the mathematics to organize a data stream by bursts of activity. If one is watching a news stream and the word Katrina suddenly appears, even if one does not understand English, one recognizes that an event has taken place somewhere in the world. The question is how do you automatically detect the sudden increase in frequency of a word and distinguish the increase from a statistical fluctuation? Jon developed a model in which bursts can be efficiently detected in a statistically meaningful manner.

A simple model for generating a sequence of events is to randomly generate the events according to a distribution where the gap x between events satisfies the distribution $p(x) = \alpha e^{-\alpha x}$. Thus, the arrival rate of events is α and the expected value of the gap between events is $\frac{1}{\alpha}$. A more sophisticated model has a set of states and state transitions. Associated with each state is an event arrival rate.

In Jon's model there is an infinite number of states $q_0, q_1, \ldots$, each having an event arrival rate. State q_0 is the base state and has the base event rate $\frac{1}{g}$. Each state q_i has a rate $\alpha_i = \frac{1}{g}s^i$ where s is a scaling parameter. In state q_i there are two transitions, one to the state q_{i+1} with higher event rate and one to q_{i-1} with lower event rate. There is a cost associated with each transition to a higher event rate state. Given a sequence of events, one finds the state sequence that most closely matches the gaps with the smallest number of state transitions.

Jon applied the methodology to several data streams demonstrating that his methodology could robustly and efficiently identify bursts and thereby provide a technique to organize the underlying content of the data streams. The data streams consisted of his own email, the papers that appeared in the professional conferences, FOCS and STOC, and finally the U.S. State of the Union Addresses from 1790 to 2002. The burst analysis of Jon's email indicated bursts in traffic when conference or proposal deadlines neared. The burst analysis of words in papers in the FOCS and STOC conferences demonstrated that the technique finds words that suddenly increased in frequency rather then finding words of high frequency over time. Most of the words indicate the emergence or sudden increase in the importance of a technical area, although some of the bursts correspond to word usage, such as the word "how" which appeared in a number of titles in the 1982 to 1988 period. The burst analysis of the U.S. State of the Union Addresses covered a 200 year time period from 1790 to 2002 and considered each word. Adjusting the scale parameter s produced short bursts of 5–10 years or longer bursts covering several decades. The bursts corresponded to national events up through 1970 at which time the frequency of bursts increased dramatically.

Table 1. Bursts in word usage in U.S. State of the Union Addresses.

energy	1812–1814	rebellion	1861–1871	Korea	1951–1954
bank	1833–1836	veterans	1925–1931	Vietnam	1951–1954
California	1848–1852	wartime	1941–1947	inflation	1971–1980
slavery	1857–1860	atomic	1947–1959	oil	1974–1981

This work on bursts demonstrated that one could use the temporal structure of data streams, such as email, click streams, or search engine queries, to organize the material as well as its content. Organizing data streams around the bursts which occur, provides us with another tool for organizing material in the information age.

Nearest neighbor

Many problems in information retrieval and clustering involve the nearest neighbor problem in a high dimensional space. A good survey of important work in this area can be found in [3]. An important algorithmic question is how to preprocess n points in d-dimensions so that given a query vector, one can find its closest neighbor. An important version of this problem is the ε-approximation nearest neighbor problem. Given a set P of vectors in d-dimensional space and a query vector x, the ε-approximation nearest neighbor problem is to find a vector y in P such that for any z in P

$$\mathrm{dist}(x, y) \leq (1 + \varepsilon)\,\mathrm{dist}(x, z).$$

Prior to Jon's work on nearest neighbor search in high dimensions [4], there was much research on this problem. Early work asked how one could preprocess P so as to be able to efficiently find the nearest neighbor to a query vector x. Most of the previous papers required query time exponential in the dimension d. Thus, if the dimension of the space was larger than $\log n$, there was no method faster than the brute force algorithm that uses time $O(dn)$. Jon developed an algorithm for the ε-approximation nearest neighbor problem that improved on the brute force algorithm for all values of d. Jon's work lead to an $O((d\log^2 d)(d+\log n))$ time algorithm for the problem [4].

The basic idea is to project the set of points P onto random lines through the origin. If a point x is closer than a point y to the query q, then with probability greater than $\frac{1}{2}$, the projection of x will be closer than the projection of y to the projection of the query q. Thus, with a sufficient number of projections, the probability that x is closer to the query than y in a majority of the projections, will be true with high probability. If

$$(1+\varepsilon)\,\mathrm{dist}(x,q) \le \mathrm{dist}(y,q),$$

then the test will fail for a majority of projections only if a majority of the lines onto which P is projected come from an exceptional set. Using a VC-dimension argument, Jon showed that the probability of more than half the lines lying in an exceptional set is vanishingly small.

Collaborative filtering

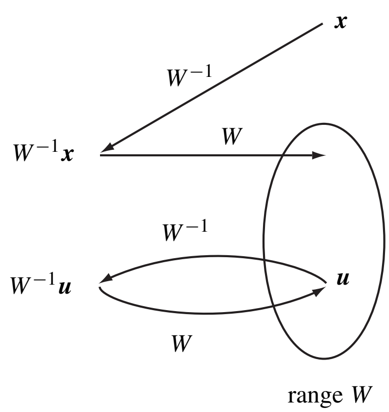
An important problem in the information age is to target a response to a person based on a small amount of information. For example, if a customer orders an item from a network store, the store may want to send him or her an advertisement based on that order. Similarly, a search engine may want to target an ad to a customer based on a query. In the case of a purchase, if for every potential item one knew the probability that the customer would buy the item, they might target an ad for the item of highest probability. However, in the case where there are possibly hundreds of thousands of items, how does one learn the probability of a customer purchasing each item based on the purchase of two or three items? If the only structure of the problem is the matrix of probabilities of customers and items, there is probably little one could do. However, if the items fall into a small number of categories and the mechanism with which a customer buys an item is that he or she first chooses a category and then having chosen a category chooses an item, one could use the structure to help in estimating the probabilities of purchasing the various items. Suppose the customer/item probability matrix is the product of a customer/category matrix times a category/item matrix. Then one can acquire information about the category/item matrix from purchases of all customers, not just the purchases of one customer.

Let A be the probability matrix of customers versus items. Then $A = PW$ where P is the probability matrix of customers versus categories and W is the probability matrix of items given a specific category. Note that the rank of A is at most the number of categories.

$$\begin{array}{c} \\ \text{customer} \end{array} \begin{array}{c} \text{item} \\ \begin{pmatrix} & A & \end{pmatrix} \end{array} = \begin{array}{c} \\ \text{customer} \end{array} \begin{array}{c} \text{category} \\ \begin{pmatrix} & \quad P \quad & \end{pmatrix} \end{array} = \begin{array}{c} \\ \text{category} \end{array} \begin{array}{c} \text{item} \\ \begin{pmatrix} & W & \end{pmatrix} \end{array}$$

Suppose we know W the matrix of probabilities of items given the categories. Let $\boldsymbol{u}$ be a row of A, the vector of probabilities with which a customer selects items. Let $\tilde{\boldsymbol{u}}$ be an estimate of $\boldsymbol{u}$ obtained from s samples. That is, the ith component of the vector $\tilde{\boldsymbol{u}}$ is $\frac{1}{s}$ times the number of times the customer selected the ith item out of s selections. The question is, how close will $\tilde{\boldsymbol{u}}$ be to $\boldsymbol{u}$? Observe that $\boldsymbol{u}$ is in the range of W and that $\tilde{\boldsymbol{u}}$ most likely is not. Thus, projecting $\tilde{\boldsymbol{u}}$ onto the range of W might improve the approximation. The question is what projection should be used? The obvious projection is to project orthogonally but this is not the only possibility.

Recall that we know W. Let W' be a generalized pseudo inverse of W. For $\boldsymbol{u}$ in the range of W (a linear combination of the columns of W) $WW'\boldsymbol{u} = \boldsymbol{u}$. However, for x not in the range of W, $WW'\boldsymbol{x}$ is obviously not $\boldsymbol{x}$ but some vector in the range of W.



Applying WW' to $\tilde{\boldsymbol{u}} - \boldsymbol{u}$ we get

$$WW'(\tilde{\boldsymbol{u}} - \boldsymbol{u}) = WW'\tilde{\boldsymbol{u}} - \boldsymbol{u}.$$

We need to bound how far the projection $WW'\tilde{\boldsymbol{u}}$ can be from $\boldsymbol{u}$. What we would like is for each component of $WW'\tilde{\boldsymbol{u}}$ to be within ε of the corresponding component of $\boldsymbol{u}$ with high probability. Then, recommending the item corresponding to the largest component would be the optimal recommendation.

If the maximum element of WW' is B, then how large can any component of $WW'\tilde{\boldsymbol{u}} - \boldsymbol{u}$ be? Stated another way, how large can $v(\tilde{\boldsymbol{u}} - \boldsymbol{u})$ be for any vector $\boldsymbol{v}$ where every element of $\boldsymbol{v}$ is bounded by some constant B? Write

$$\tilde{\boldsymbol{u}} = \frac{1}{s}\sum_{i=1}^{s} \bar{\boldsymbol{u}}_i$$

where $\bar{\boldsymbol{u}}_i$ is the indicator vector for the ith selection. Then $\boldsymbol{v}^T\tilde{\boldsymbol{u}} = \frac{1}{s}\sum_{i=1}^{s}\boldsymbol{v}^T\bar{\boldsymbol{u}}_i$. The terms in the summation are independent random variables since the selections are independent. The variance of the product of an element of v times an element of $\bar{\boldsymbol{u}}_i$ is at most $\left(\frac{B}{s}\right)^2$, and hence the variance of $\boldsymbol{v}^T\tilde{\boldsymbol{u}}$ is at most $\frac{B^2}{s}$. (Elements of $\bar{\boldsymbol{u}}_i$ have value 0 or $\frac{1}{s}$ and the maximum of v is B.) Hence, by Chebyshev's inequality, the probability that $\boldsymbol{v}^T\tilde{\boldsymbol{u}}$ will differ from its expected value by more than ε is bounded.

$$\text{Prob}(|\boldsymbol{v}^T\tilde{\boldsymbol{u}} - \boldsymbol{v}^T\boldsymbol{u}| < \frac{B^2}{\varepsilon^2 s}.$$

The above result tells us that $\boldsymbol{v}^T\tilde{\boldsymbol{u}}$ will be close to its expected value of $\boldsymbol{v}^T\boldsymbol{u}$ provided no element of $\boldsymbol{v}$ is excessively large. Thus, in projecting $\tilde{\boldsymbol{u}}$ onto the space of W we want to use a projection WW' where W' is selected so that it has no excessively large element.

This led Jon and his colleague Mark Sandler [8] to use linear programming to find a pseudo inverse in which the maximum element was bounded by $\frac{1}{\Gamma}$ where $\Gamma = \min_{\|x\|_1=1} \|W\boldsymbol{x}\|_1$. This lead to a collaborative filtering algorithm that recommends an item whose probability of being purchased by the customer is within an ε of the highest probability item with high probability. What is so important about this work is that it can be viewed as the start of a theory based on the 1-norm. Although much of the theory of approximation is based on the 2-norm and in fact approximating a matrix A by a low rank matrix A_k one can prove that the Frobenius norm of the error matrix is minimized by techniques based on the 2-norm, the error is not uniformly distributed. Furthermore, the error is strongly influenced by outliers. Use of the 1-norm is a promising approach to these problems.

Closing remarks

This brief summary covers five important research thrusts that are representative of Kleinberg's work. His web page contains many other exciting results of which I will mention three. First is his early work with Eva Tardos [9] on network routing and the disjoint paths problem. They developed a constant-factor approximation algorithm for the maximum disjoint paths problem in the two-dimensional grid graph. Given a designated set of terminal node pairs, one wants to connect as many pairs as possible by paths that are disjoint. Their algorithm extends to a larger class of graphs that generalizes the grid.

Second is his early work with Borodin, Ragahavan, Sudan and Williamson [1] on the worst-case analysis of packet-routing networks. This work presents a framework for analyzing the stability of packet-routing networks in a worst-case model without probabilistic assumptions. Here one assumes packets are injected into the network, limited only by simple deterministic rate bounds, and then shows that certain standard protocols guarantee queues remain bounded forever while other standard protocols do not.

Third is his work with Eva Tardos on classification with pairwise relationships [10]. This paper gives an algorithm for classification in the following setting: We are given a set of objects (e.g. web pages) to classify, each into one of k different types, and we have both local information about each object, as well as link information specifying that certain pairs of objects are likely to have similar types. For example, in classifying web pages by topic (or into some other categories), one may have an estimate for each page in isolation, and also know that pairs of pages joined by hyperlinks are more likely to be about similar topics.

Conclusions

Jon's work has laid a foundation for the science base necessary to support the information age. Not only is the work foundational mathematically but it has contributed to the economic growth of industries.

References

[1] Borodin, A., Kleinberg, J., Raghavan, P., Sudan, M., and Williamson, D., Adversarial queueing theory. In *Proceedings of the 28nd Annual Symposium on Theory of Computing*, 1996, 376–385.

[2] Brin, S., and Page, L., Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proceedings of the 7th International World Wide Web Conference*, 1998, 107–117.

[3] Indyk, P., Nearest Neighbors in High-dimensional Spaces. In *Handbook of Discrete and Computational Geometry* (J. E. Goodman and J. O'Rourke, eds.), 2nd edition, CRC Press, Boca Raton, FL, 2004, 877–892.

[4] Kleinberg, J., Two algorithms for nearest-neighbor search in high dimensions. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, ACM Press, New York 1997, 599–608.

[5] Kleinberg, J., Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, ACM, New York 1998, 668–677.

[6] Kleinberg, J., The small-world phenomenon: An algorithmic perspective. In *Proceedings of the 32nd Annual Symposium on Theory of Computing*, ACM Press, New York 2000, 163–170.

[7] Kleinberg, J., Bursty and Hierarchical Structure in Streams. In *Proceedings of the 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, ACM, New York 2002, 91–101.

[8] Kleinberg, J., and Sandler, M., Using Mixture Models for Collaborative Filtering. In *Proceedings of the 36th ACM Symposium on Theory of Computing*, ACM, New York 2004, 569–578.

[9] Kleinberg, J., and Tardos, E., Disjoint paths in densely embedded graphs. In *Proceedings of the 36th IEEE Symposium on Foundations of Computer Science*, IEEE Comput. Soc. Press, Los Alamitos, CA, 1995, 52–59.

[10] Kleinberg, J., and Tardos, E., Approximation Algorithms for Classification Problems with Pairwise Relationships: Metric Labeling and Markov Random Fields. In *Proceedings of 40th IEEE Symposium on Foundations of Computer Science*, IEEE Comput. Soc. Press, Los Alamitos, CA, 1999, 14–23.

[11] Liben-Nowell, D., Novak, J., Kumar, R., Raghavan, P., and Tomkins, A., Geographic routing in social networks. *Proc. Natl. Acad. Sci. USA* **102** (2005), 11623–11628.

[12] Milgram, S., The small world problem. *Psychology Today* **1** (1967), 60–67.

[13] Salton, G., *Automatic Text Processing.* Addison-Wesley, Reading, Mass., 1989.

[14] Watts, D., and Strogatz, S., Collective dynamics of small-world networks. *Nature* **393** (1998), 440–442.

Department of Computer Science, Cornell University, Ithaca, New York 14953, U.S.A.