Four Results of Jon Kleinberg

A Talk for St.Petersburg Mathematical Society

Yury Lifshits

Steklov Institute of Mathematics at St.Petersburg

May 2007

Outline

Part I

History of Nevanlinna Prize

Career of Jon Kleinberg

Nevanlinna Prize

The Rolf Nevanlinna Prize is awarded **once every 4 years** at the International Congress of Mathematicians, for outstanding contributions in **Mathematical Aspects of Information Sciences** including:

1. All mathematical aspects of computer science, including complexity theory, logic of programming languages, analysis of algorithms, cryptography, computer vision, pattern recognition, information processing and modelling of intelligence.
2. Scientific computing and numerical analysis. Computational aspects of optimization and control theory. Computer algebra.

Only scientists **under 40** are eligible

Previous Winners

1982 Robert Tarjan: data structures, graph algorithms

1986 Leslie Valiant: learning theory, complexity, parallel computing

1990 Alexander Razborov: work around P vs. NP

1994 Avi Wigderson: complexity and cryptography

1998 Peter Shor: quantum algorithm for factoring problem

2002 Madhu Sudan: coding theory, probabilistically checkable proofs and inapproximability

Short Bio of Jon Kleinberg

1971 Jon Kleinberg was born in Boston

1993 Bachelor degree from Cornell

1996 Ph.D. from MIT (advisor Michel X. Goemans)

Since 1996 Cornell faculty

2006 Nevanlinna Prize

More about Jon Kleinberg

- According to DBLP: 108 papers and 85 coauthors for 1992-2006
- H-Index = 36 (according to scholar.google.com)
- Book "Algorithm Design"(2005, with Éva Tardos)
- NSF Career Award, ONR Young Investigator Award, MacArthur Foundation Fellowship, Packard Foundation Fellowship, Sloan Foundation Fellowship, "Faculty of the Year" Cornell'2002
- Strong connections to IBM Almaden Research Center
- Courses "The Structure of Information Networks" and "Randomized and High-Dimensional Algorithms"
- Chair of STOC'06

Research Style of Jon Kleinberg

- **Direction:** from practical problems to mathematical ideas
- **Motivation:** make life better
- **Validation:** mathematical proofs **and** experiments
- **Connections with:** sociology
- **Key component:** new models/formalizations, not proofs

Part II

Authoritative sources in a hyperlinked environment
Jon Kleinberg — SODA'98

2580 citations
according to scholar.google.com, May 2007

Challenge

How to define the most relevant webpage to "Bill Gates"?

Naive ideas

- By frequency of query words in a webpage
- By number of links from other **relevant** pages

Web Search: Formal Settings

- Every webpage is represented as a weighted set of keywords
- There are hyperlinks (directed edges) between webpages

Conceptual problem: define a relevance rank based on keyword weights and link structure of the web

HITS Algorithm

1. Given a query construct a **focused subgraph** $F(query)$ of the web
2. Compute **hubs and authorities** ranks for all vertices in $F(query)$

Focused subgraph: pages with highest weights of query words **and** pages hyperlinked with them

Hubs and Authorities

Mutual reinforcing relationship:

- A good **hub** is a webpage with many links **to** query-authoritative pages
- A good **authority** is a webpage with many links **from** query-related hubs

Hubs and Authorities: Equations

$$a(p) \sim \sum_{q:(q,p)\in E} h(q)$$

$$h(p) \sim \sum_{q:(p,q)\in E} a(q)$$

Hubs and Authorities: Solution

Initial estimate:

$$\forall p : a_0(p) = 1, h_0(p) = 1$$

Iteration:

$$a_{k+1}(p) = \sum_{q:(q,p)\in E} h_k(q)$$

$$h_{k+1}(p) = \sum_{q:(p,q)\in E} a_k(q)$$

We normalize $\bar{a}_k, \bar{h}_k$ after every step

Convergence Theorem

Theorem

Let M be the adjacency matrix of focused subgraph $F(query)$. Then $\bar{a}_k$ converges to principal eigenvector of $M^T M$ and $\bar{h}_k$ converges to principal eigenvector of MM^T

Lessons from Hubs and Authorities

- Link structure is useful for relevance sorting
- Link popularity is defined by linear equations
- Solution can be computed by iterative algorithm

Part III

Two algorithms for nearest-neighbor search
in high dimensions
Jon Kleinberg — STOC'97

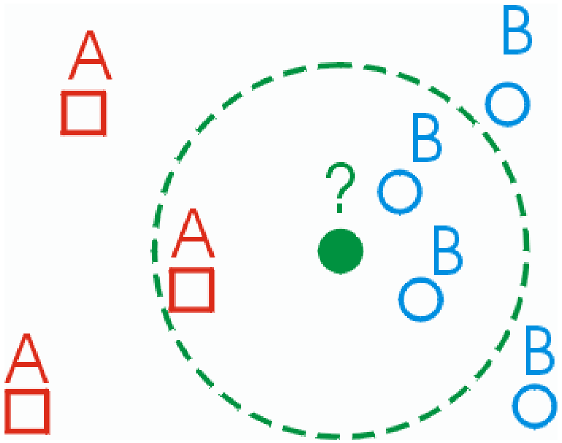
173 citations
according to scholar.google.com, May 2007

Informal Problem Statement

To preprocess a database of *n* objects
so that given a query object,
one can effectively determine
its nearest neighbors in database

First Application (1960s)

Nearest neighbors for classification:



Picture from http://cgm.cs.mcgill.ca/ soss/cs644/projects/perrier/Image25.gif

Applications

What applications of nearest neighbors do you know?

- Statistical data analysis, e.g. medicine diagnosis
- Pattern recognition, e.g. for handwriting
- Code plagiarism detection
- Coding theory
- Future applications: recommendation systems, ads distribution, personalized news aggregation

Challenge

Brute force algorithm
No preprocessing
$\mathcal{O}(nd)$ query time
for n points in d-dimensional space

Open Problem: Is there any preprocessing method with data structure of $poly(n+d)$ size and $o(nd)$ query time?

Approximate Nearest Neighbors

Definition
p is ε-approximate nearest neighbor for q
iff $\forall p' \in DB: \qquad d(p,q) \leq (1+\varepsilon)d(p',q)$

Kleinberg Algorithm

Theorem

For every ε, δ there exists a data structure with $\mathcal{O}^*(d^2 n)$ construction time and $\mathcal{O}(n + d \log^3 n)$ query processing time. It correctly answer ε-nearest neighbor queries with probability $1-\delta$.

Data Structure Construction

1. Choose $l = d \log^2 n \log^2 d$ random vectors $V = \{v_1, \ldots, v_l\}$ with unit norm
2. Precompute all scalar products between database points and vectors from V

Random Projection Test

Input: points x,y and q, vectors $u_1, \ldots, u_k$
Question: what is smaller $|x - q|$ or $|y - q|$?

Test:
For all i compare $(x \cdot v_i - q \cdot v_i)$ with $(y \cdot v_i - q \cdot v_i)$
Return the point which has "smaller"
on majority of vectors

Query Processing

1. Choose a random subset Γ of V, $|\Gamma| = \log^3 n$
2. Compute scalar products between query point q and vectors from Γ
3. Make a tournament for choosing a nearest neighbor:
   1. Draw a binary tree of height $\log n$
   2. Assign all database points to leafs
   3. For every internal point (say, x vs. y) make a random projection test using some vectors from Γ

Part IV

The small-world phenomenon:
An algorithmic perspective
Jon Kleinberg — STOC'00

433 citations
according to scholar.google.com, May 2007

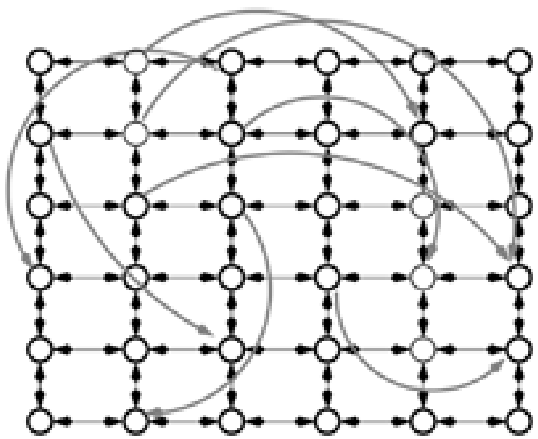
Milgram's Small World Experiment

1. Starting point: Wichita/Omaha, endpoint: Boston
2. **Basic information** about a target contact person in Boston was initially sent to randomly selected individuals.
3. If recipient knew the contact person, he/she should **forward** the letter **directly** to that person
4. If recipient did not personally know the target then he/she should **forward** the package **to a friend** or relative they know personally that is more likely to know the target
5. When and if the package eventually reached the contact person in Boston, the researchers count the number of times it had been forwarded from person to person.

Small World Model

- $n \times n$ grid of n^2 nodes
- Every node p is connected to its direct neighbors: right, left, up and down
- Additionally, every node p has an arc to a "random" node q, where probability for q to be chosen is proportional to $|p - q|^{-\alpha}$, $\alpha \geq 0$

Small World Model



Picture from `www.math.cornell.edu/~durrett/smw/kleinberg2.gif`

Navigability

A graph is **navigable**, if there exists **decentralized** algorithm finding connecting paths in $polylog(n)$ time

Whether small world is navigable?

Kleinberg's Results

Theorem

For $\alpha = 2$ small world is navigable, for all other nonnegative values of α it is not.

Part V

Bursty and Hierarchical Structure in Streams
Jon Kleinberg — KDD'02

150 citations
according to scholar.google.com, May 2007

Streams and Bursts

- A stream of events
- Every event = set of keywords + time stamp

How should we identify time intervals with unusually high frequency of a specific keyword?

Conceptual Solution

Hidden Markov Model methodology:

- There is a "creature" who generates our stream
- This creature can be described as a finite automaton of known structure but with unknown state sequence
- We will find "the most fitting" sequence of states for our data
- Based on this sequence we can identify all bursts

Very Simple Example (1/2)

Keyword: "grant"
Events: every day either there is an email with this keyword or there is not

Example Data: we have email archive for two weeks

01110100001000

Very Simple Example (1/2)

01110100001000

Automaton: two states "grant deadline" and "vacations"

Fitting function: 1 point penalty for mismatches "grant deadline — no grant emails" and "vacations — email with grants", 1 point penalty for switching state of automaton

Optimal sequence: VDDDDDV VVVVVVV

Algorithm for Detecting Bursts

How to compute the optimal state sequence?

Dynamic programming:

- For every day d and every state s we will compute the optimal state sequence for period $[1..d]$ ending with state s
- When a data for new day comes we try all values for yesterday and choose the best one
- For optimal sequence for the whole interval $[1..D]$ we just take the maximum over all states

Home problem

Find an anagram for "KLEINBERG"

Highlights

- Hubs and Authorities is an iterative algorithm for computing relevance rank
- Small world always can have small diameter but no decentralized method for finding short paths
- Bursts can be identified as states of imaginary automaton that generates event stream
- Nearest neighbors can be found by looking at projections to random vectors

Thank you for your attention!
Questions?

References

All materials of this talk will be published at **my homepage**: http://logic.pdmi.ras.ru/~yura

Jon Kleinberg
Authoritative sources in a hyperlinked environment — SODA'98
http://www.cs.cornell.edu/home/kleinber/auth.pdf

Jon Kleinberg
The small-world phenomenon: An algorithmic perspective — STOC'00
http://www.cs.cornell.edu/home/kleinber/swn.ps

Jon Kleinberg
Bursty and Hierarchical Structure in Streams — KDD'02
http://www.cs.cornell.edu/home/kleinber/bhs.ps

Jon Kleinberg
Two algorithms for nearest-neighbor search in high dimensions — STOC'97
http://www.cs.cornell.edu/home/kleinber/stoc97-nn.pdf

Relevant Links

Official site of Nevanlinna Prize
http://www.mathunion.org/Prizes/Nevanlinna/index.html

Homepage of Jon Kleinberg
http://www.cs.cornell.edu/home/kleinber/

Jon Kleinberg at DBLP
http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/k/Kleinberg:Jon_M=.html

IMU news release
http://www.mathunion.org/medals/2006/kleinbergENG.pdf

Interview of Jon Kleinberg to "Technology Research News"
http://www.trnmag.com/Stories/2005/120505/View_Jon_Kleinberg_120505.html

A talk by Jon Kleinberg on Yahoo! Video
http://video.yahoo.com/video/play?vid=62055