Searching Without Meaning

Daniel Lemire,

Téluq, Université du Québec à Montréal (UQAM)

National Research Council Canada
Conseil national de recherches Canada

Why care about search?

Increasingly, search is our mechanism for how we understand ourselves, our world, and our place within it. It's how we navigate the one infinite resource that drives human culture : knowledge. (John Batelle, The Search)

Some basic definitions

- A "vector" is just an array of n values $(0.1, 3.4, \ldots, 4.2)$
- You can represent a n-dimensional space as a line going from $(0, \ldots, 0)$ to $(0.1, 3.4, \ldots, 4.2)$
- You can talk about the "angle" between two vectors (defined by the scalar product!)

The fundamental problem

- The answer is there somewhere
- It is even in digital format
- but where?

A researcher's nightmare!

- Someone must have worked on finding "frequent words" in a text
- What are the good algorithms?
- Turns out that to find the answer, you have to search for "frequent strings"!!!

Historical Perspective

- 1945 death of Hitler?
- Electronic Computers follow soon after
- Early application: finding the document we need (replacing librarian)
- 50 years later we still have librarians, but Google is hurting them!

What people thought the solution would be

- Issue a query “Hitler dead”
- Semantic Feedback: did you mean “Hitler is dead” or “dead because of Hilter”
- Fundamental idea: ask user to choose from a controlled set of answers!

The Semantic Web idea

- How to find the information I need on the web?
- Berners-Lee, in early nineties, had imagined semantic solutions!
- Became the Semantic Web, based on RDF, ontologies and so on.
- Net result is similar to semantic feedback: at some point, the user must make a semantic choice!

What works: vector model

- Jean est à l'usine. Pierre est aussi à l'usine. (document 1)
- Jean mange des pommes. (document 2)
- Pierre est à la ferme. (document 3)

Cooccurence matrix (term frequency)

term	document 1	document 2	document 3
Jean	1	1	0
ferme	0	0	1
usine	2	0	0
pommes	0	1	0
Pierre	1	0	1

Rare words are more important

- Frequent words like “the” don’t matter!
- Measure the “usefulness” of a term with inverse document frequency
- $idf = \log |D|/df$

Rare words are more important

term	idf
Jean	0.6
ferme	1.6
usine	1.6
pommes	1.6
Pierre	0.6

TF-IDF matrix

term	document 1	document 2	document 3
Jean	0.6	0.6	0
ferme	0	0	1.6
usine	3.2	0	0
pommes	0	1.6	0
Pierre	0.6	0	0.6

How to query a vectorial model?

- ▷ query like “usine pommes” become a vector $0, 0, 1, 1, 0$
- ▷ correct for idf, $0, 0, 1.6, 1.6, 0$
- ▷ Compute angle document x versus my query
- ▷ pick smallest angle!

What works: Google

- Uses the web topology to recommend web pages!
- If people are interested enough to link...
- Maybe you'll like it....
- **Reference**: The PageRank Citation Ranking: Bringing Order to the Web by Lawrence Page, Sergey Brin, Rajeev Motwani, & Terry Winograd - 1999

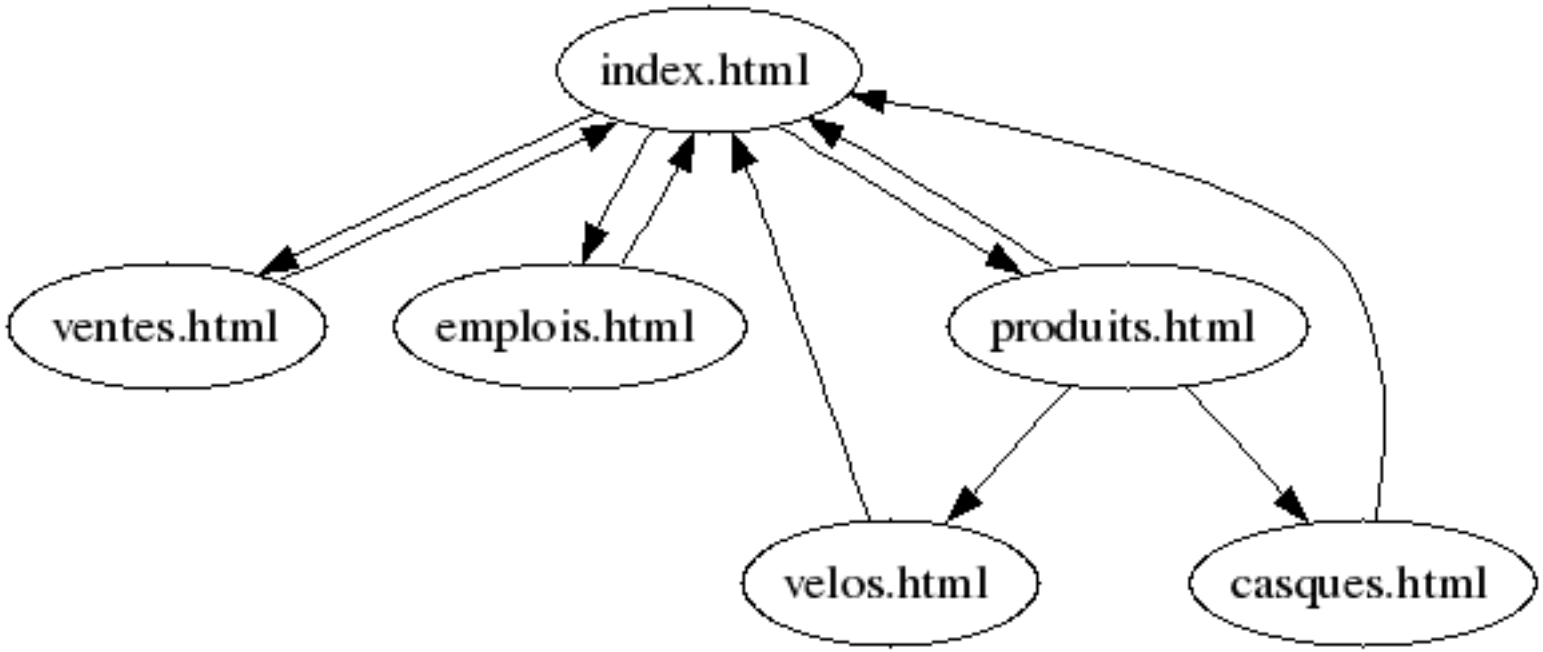
PageRank is...

You jump from one page to another

Over time, how likely are you to be visiting a page x?

That's the PageRank!!!

Random-Surfer model



Markov process matrix

$$\begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Markov process convergence

$$\begin{bmatrix} x_1 \ldots x_5 \end{bmatrix} \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{10000}$$

PageRank with restart

Some part of the web are “sinks”: they are linked to, but don’t “link out”

To compensate, have the random surfer restart from a random page from time to time (with probability 0.15)

Markov process convergence ($q = 0.85$)

$$\begin{bmatrix}
\frac{1-q}{6} & \frac{q}{3}+\frac{1-q}{6} & \frac{q}{3}+\frac{1-q}{6} & \frac{q}{3}+\frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} \\
\frac{q}{3}+\frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} & \frac{q}{3}+\frac{1-q}{6} & \frac{q}{3}+\frac{1-q}{6} \\
q+\frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} \\
q+\frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} \\
q+\frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} \\
q+\frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6} & \frac{1-q}{6}
\end{bmatrix}$$

What works: Amazon

- "Users who bought this, also bought this..."
- Compute cosine between item vectors
- **Reference**: Amazon.com Recommendations: Item-to-Item Collaborative Filtering by Greg Linden, Brent Smith, and Jeremy York in IEEE Internet Computing, Feb. 2003.

Purchase stats

term	Sex toy	dog	cat	meat
Einstein	0	0	1	0
Hitler	1	1	0	1
Mom	0	1	0	0

People who bought sex toys also bought meat

Millions of MP3s...

- We can download *millions* of MP3s
- legally («indie music»)
- However, it isn't done a lot
- Why? Difficult to find what we like!
- Hard to replace the «distribution channels»

Example of our work: inDiscover

- Music recommender site:

  http://www.indiscover.net

- Indie Musicians submit their music (MP3), add metadata

- Multidimensional ratings, Slope One, Inference Engine

- Hundreds of great songs... find what **you** link, build taylored playlists

Why do we want to gather subjective metadata?

You are in a video club. You are looking for a movie.

What genre of movie? A comedy, produced after 1999, in Virginia and with a 2 millions+ budget?

No. You want a movie you, and people like you, will like.

The same thing applies to books, courses, web sites...

Taxonomies only go so far

Metadata as a solution?

People lie, people are lazy, people are stupid. - Cory Doctorow, Metacrap: Putting the torch to seven straw-men of the meta-utopia

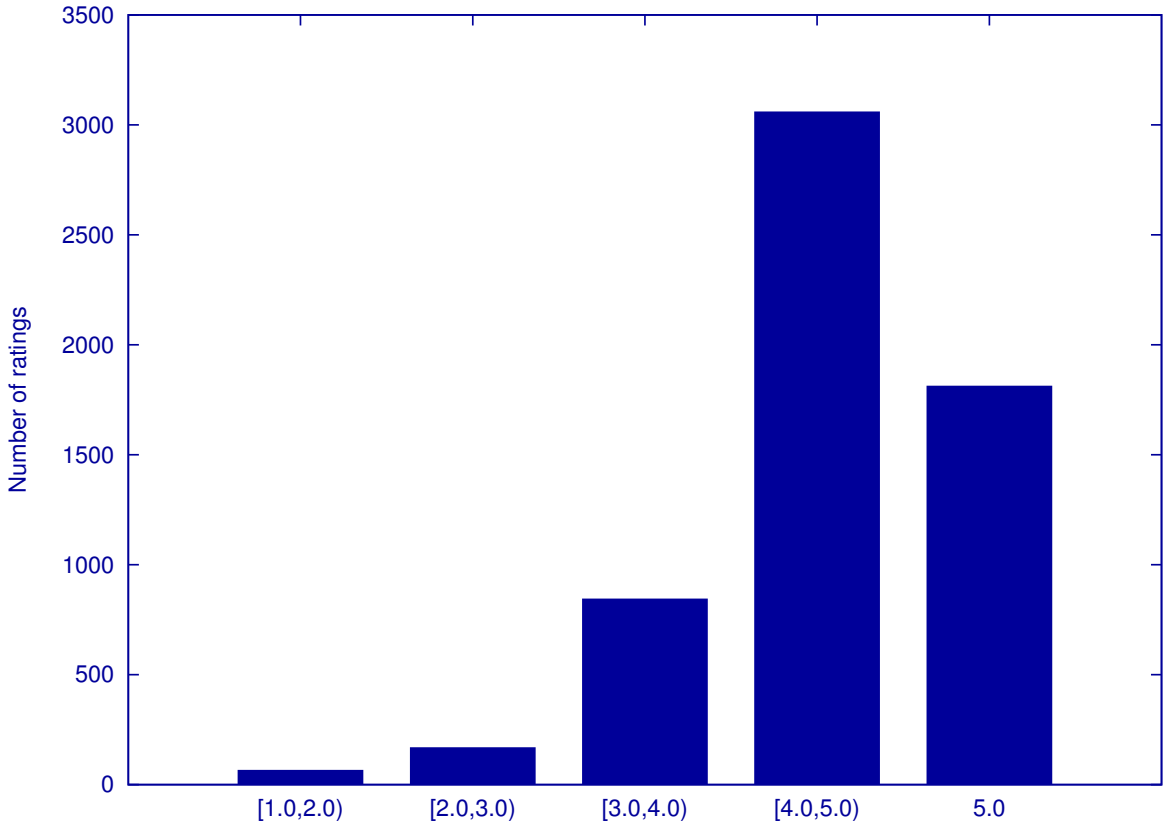
Even Semantic Web People...

Efficient rating of Data, Metadata, and Raters is essential for Semantic Subwebs that want to compete with good, old paper-based libraries. - Harold Boley, The Semantic Web in Ten Passages

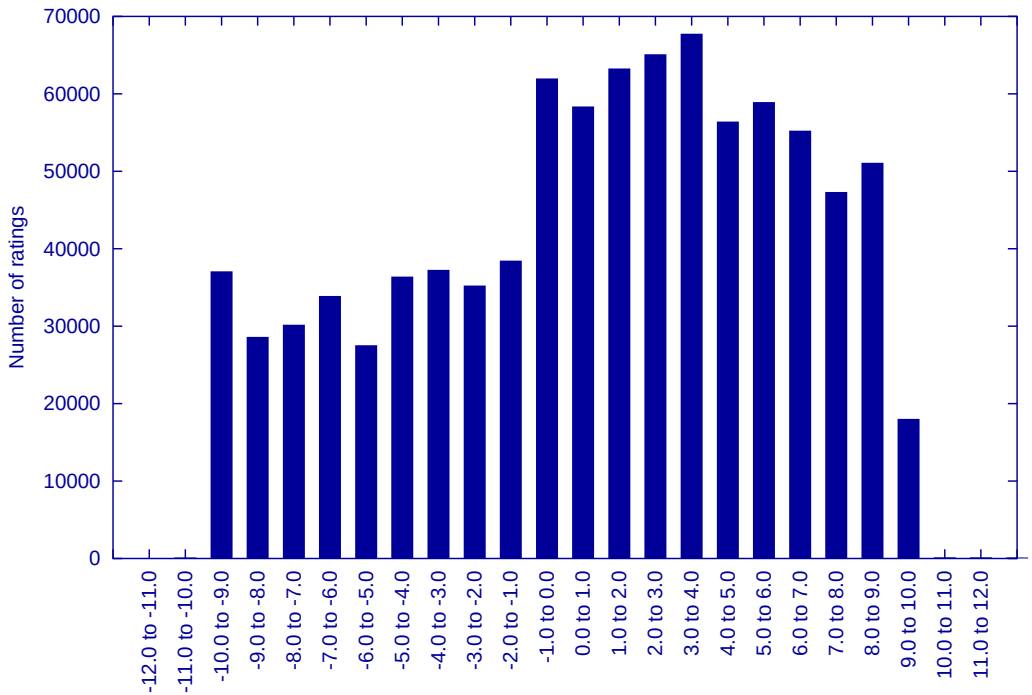
Rating-Based Collaborative Filtering

- We ask people to “rate” an item
- Example: students at the end of a course
- Example: Firefly (web site), ePinions, inDiscover
- Objective: personalized recommendations

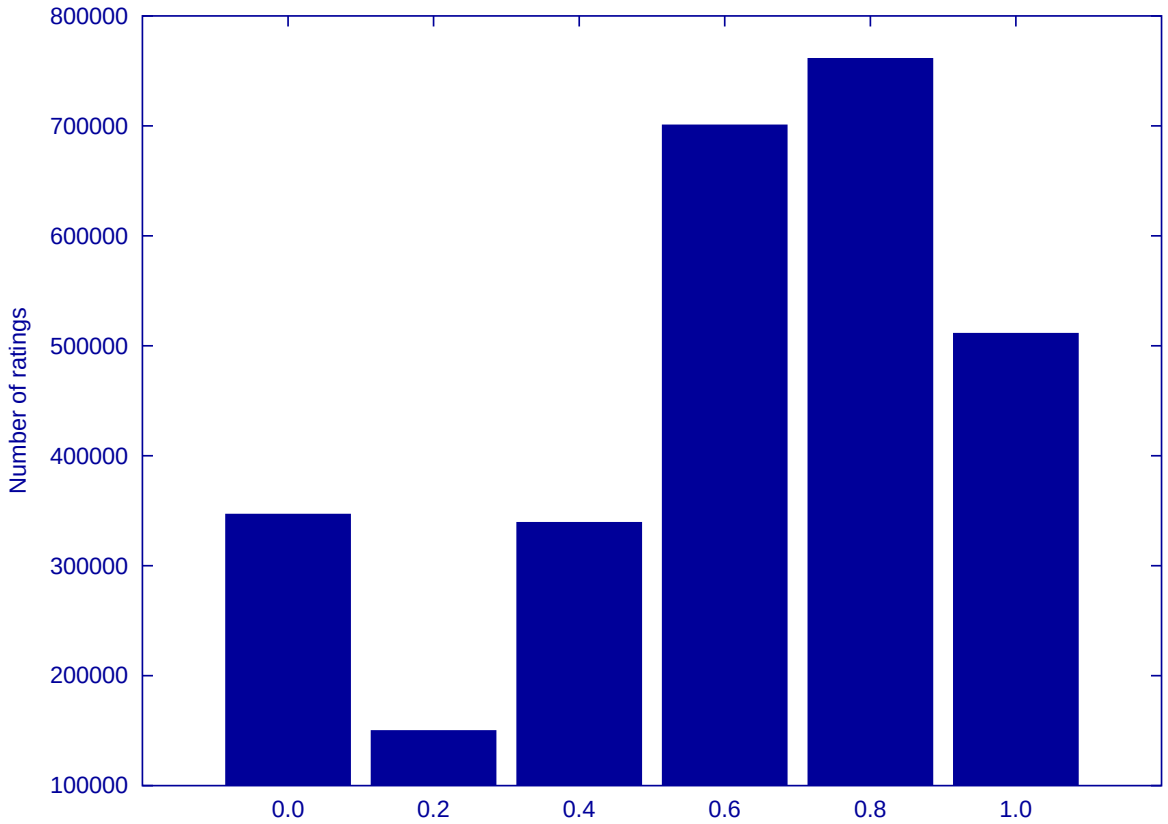
How do Amazon.com users rate?



How do Jester users rate?



How do EachMovie users rate?



Conjecture

- Words in text is distributed according to a power law. (Zipf and Mandelbrot laws)
- Ratings are distributed according to lognormal (?) law.

What to do with ratings?

- «Gold Balling»: what are the best items
- «Black Balling»: what are the worse items
- «White Balling»: constraint-based searches (multidimensional?)
- Hybrid models: ratings + content analysis + ontologies + ...

Status

- ▷ First collaborative filtering papers ($\simeq 1994$)
- ▷ Harder techniques (metrics, «Machine Learning», etc.) ($\simeq 1998$)
- ▷ Not much progress accuracy-wise (reached a "ceiling") ($\simeq 2001$)
- ▷ Very little theory, no journal, no dedicated conference

Evaluation

- An "evaluation" u is a sparse vector containing the ratings given by a user to some items.
- One evaluation, one user
- (a function $u : S(u) \rightarrow \mathbb{R}$ where $S(u) \subset \mathfrak{1}$)
- An evaluation is very sparse
- Amazon sells millions of books, how many have you rated?

Careful

- Unlike a vector model in IR, unknown values are not zeroes.
- The set of evaluations is not an Hilbert or Banach space, not even vector space!
- Given $u = (?, 3, 4)$ and $v = (2, ?, 4)$...
  - What is $u + v$? (wrong answer: $(2, 3, 8)$)

Predictors, quality measures

We can measure the accuracy using «All But 1» (leave-one-out)

Take $u = (4,4,5)$, kill last rating $u' = (4,4,?)$

Compute $\left|P(u')_3 - u_3\right|$!

Necessary conditions

Independance with respect to «normalization»

Ratings are written out as numbers...

Whether the rating scale is **-10–10** or **-1–1** or **0–1**

That's arbitrary!!!

Stability is a nice property

Should be fake-ratings resilient

(Google has to cope with fake links)

Things to avoid: N-nearest neighbours for N small.

Predictor Types

- Memory-based predictors: slow, but reliable
- Model-based predictors: faster, Machine Learning

Memory-based predictors

Memory-based predictors seek similar users and compute a weighted sum...

$$P(u)_i = \frac{\sum_{v \in S_i(\chi))} \gamma(u,v) v_i}{\sum_{v \in S_i(\chi))} |\gamma(u,v)|}.$$

where γ is a similarity measure like the Pearson correlation .

Model-Based Predictors

Use the Data to Learn. Wide range: Bayes, Clustering, PCA, etc.

Collaborative Filtering $\neq$ *Machine Learning*

Collaborative filtering algorithms must be:

- ▷ easy to implement and to manage
- ▷ easy to update
- ▷ very fast
- ▷ require little from new visitors
- ▷ accurate, but recall accuracy ceiling...

«Item-Item» or «Item-to-Item» or «Item-Based»

- Item-based (Vucetic, Obradovic 2000),(Sarwar, Karypis et al. 2001)
- Easy to implement, scalable
- Simple trick: for each pair of items i, j, solve for f such that f predicts ratings on j given rating on j

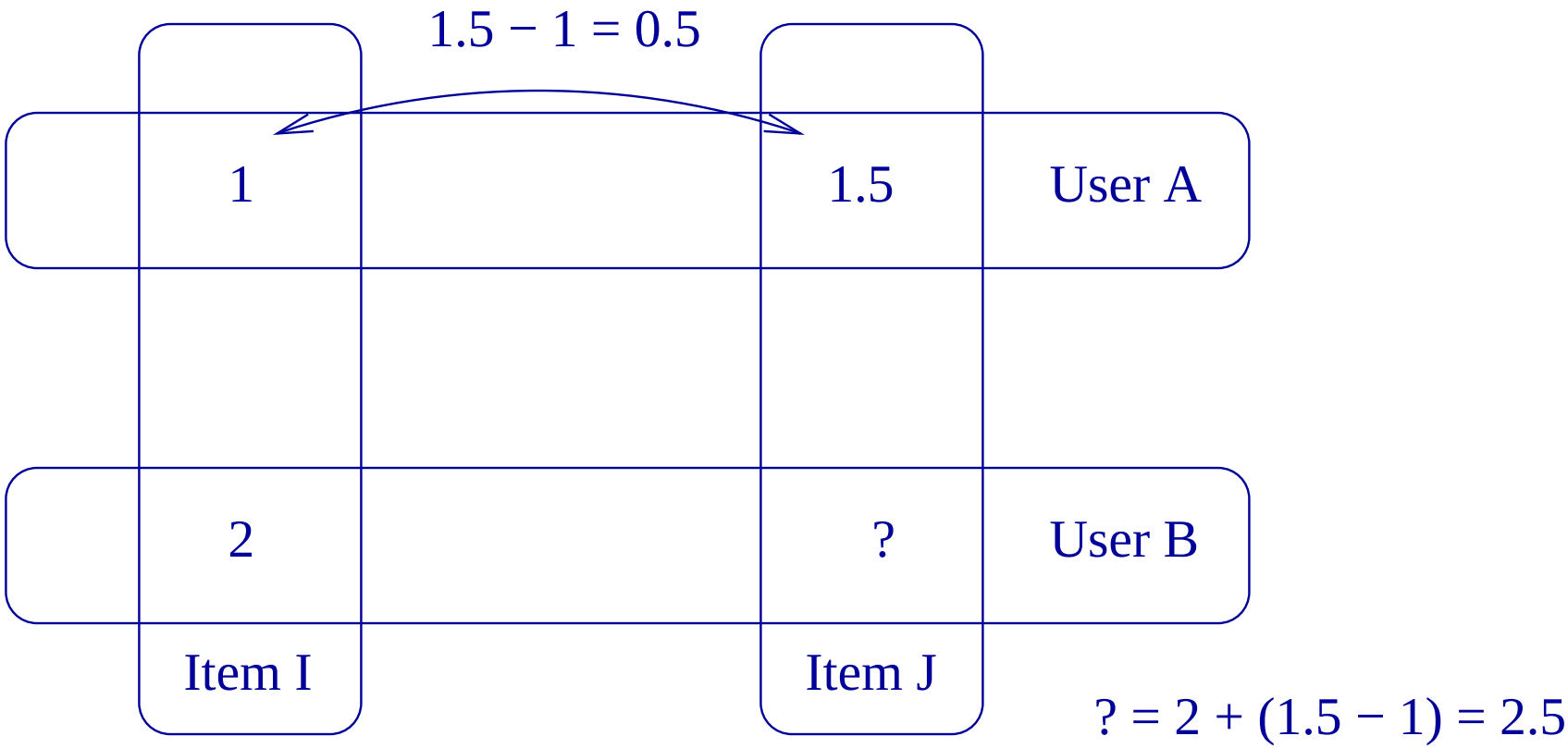
How to choose f?

- ▷ Other researchers went with $f(x) = ax + b$: where a, b are found by regression
- ▷ Binary case (bought or not) investigated by Amazon (Linden et al. 2004) and Verizon (Demiriz 2004)

Slope One Model

- The Slope One model is $f(x) = x + b$ (by opposition to $f(x) = ax + b$ or $f(x) = ax^2 + bx + c$
- You reduce storage... and improve accuracy by a lot!
- And easier to understand: b is average diff. between i abd j

Slope One Model:diagram



The Slope One Secret

- The secret keyword is “overfitting”!
- Always choose the simplest model that will do the job!
- Slope One is pretty much it!

Glue predictions together

- If user rated x items, we must combine x predictions
- We tried several thigns: averaging, weighted averaging by number of coratings (i, j)
- Weighted averages works best.

Performance

- Item-based model is great when you have few items: small item-item matrix
- Also works with large matrices (think Amazon), but requires sparse matrices
- Can make matrices sparser by setting lesser weights to zero ("Icebergs")
- Our implementation scales up enough for Bell Canada to buy it!

Summary

- Semantic is not very useful for search, at least so far
- Vector-based model, Web-page topology and purchase patterns are most useful
- Ratings might work too?