

Top- k Processing for Search and Information Discovery in Social Applications

Lecture 5: Open Problems

Sihem Amer-Yahia



Julia Stoyanovich



Social top- k @ Joint RuSSIR/EDBT Summer School 2011

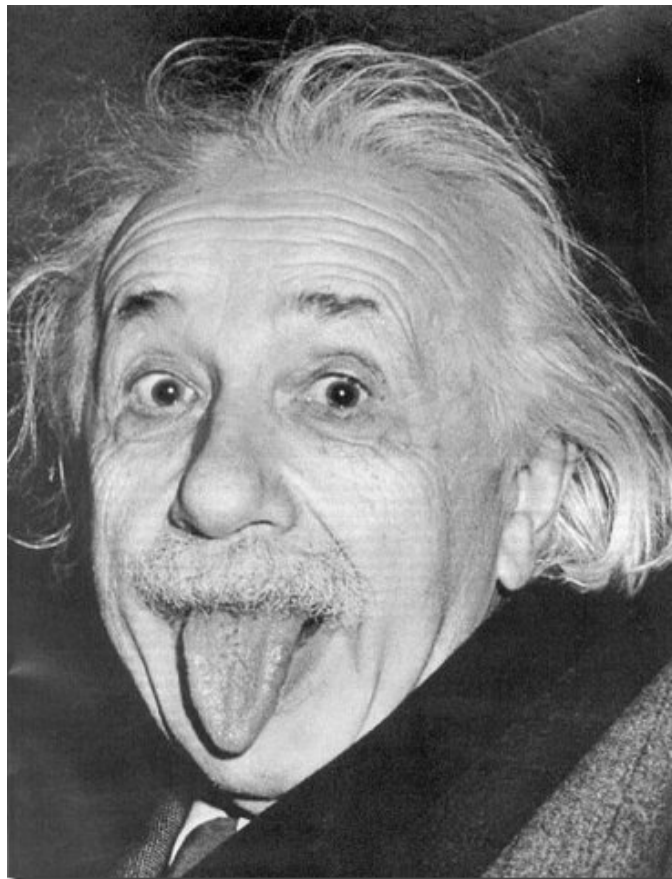
Summary of last lectures

- **Fundamental algorithms**
 - Use the inverted list indexing structure
 - Have an access strategy and a stopping condition
 - TA and NRA
 - Applicable to social applications (network-aware search and group recommendation)
- **User studies**
 - necessary to verify user satisfaction given a ranking semantics
 - large-scale tools such as Amazon Mechanical Turk

Quote of the day

Imagination is more important than knowledge.

~Albert Einstein



Novel top- k challenges

- **Traditional top- k processing**
 - a single scoring function used to rank items
 - highest scoring items are returned
- **Beyond traditional top- k**
 - what if the items we are returning are composite, i.e., formed by many other items?
 - e.g., composition based on peer pressure/co-purchasing/co-reviewing
 - goal: compute a ranked list of composite items satisfying a budget (price, time)
 - what if rank aggregation relies on multiple scoring semantics?
 - e.g., ranking semantics may reflect different movie reviewer populations
 - goal: compute multiple ranked lists, one for each semantics
 - what if desirable items cannot be captured with a single score?
 - e.g., *diversity* may reflect preferences of different subsets of friends of a user
 - goal: find most diverse set of items s.t. their individual score does not drop below a given threshold

Outline

✓ Intro

- **Composite top- k**
 - Problem definition
 - Overview of a solution
- **Multi-top- k**
- **Diverse top- k**
- **Concluding thoughts**

Composite item



Item bundles on Amazon

Frequently Bought Together



Total List Price: \$45.94

Price For All Three: **\$31.77**

 [Add all three to Cart](#) [Add all three to Wish List](#)

[Show availability and shipping details](#)

- ✓ **This item:** The Harafish by Naguib Mahfouz
- ✓ [Children of the Alley: A Novel](#) by Peter Christopher Theroux
- ✓ [The Yacoubian Building: A Novel](#) by Humphrey T. Davies

Complementary items: JCPenney

Stafford® Essentials Single-Button Tuxedo Coat




\$90.00 to \$100.00

Original \$180.00 to \$200.00

"Why rent when you can own?"

- Single-button front or three-button front coat
- Satin lapel
- Natural shoulders
- 100% worsted wool
- Satin polyester lapel

 you might also like



Enamel Inlay Cuff Links and Stud Set

\$24.99

Orig. \$30.00



Stafford® Essentials Tuxedo Vest

\$29.99

Orig. \$60.00



Engravable Oval Cuff Links

\$29.99

Orig. \$35.00



Stafford® Essentials Tuxedo Shirt Set

\$29.99

Orig. \$60.00



Stafford® Essentials Pleated Tuxedo Pant

\$39.99 - 49.99

Orig. \$80.00 - 100.00

JCPenney (budget = \$175)

The diagram illustrates two different tuxedo outfit configurations, each centered around a 'Stafford® Essentials Single-Button Tuxedo Coat' priced at \$90. Purple lines connect the central coat to its various accessories.

Left Configuration:

- Stafford® Essentials Tuxedo Vest:** \$29.99
- Stafford® Essentials Tuxedo Shirt Set:** \$29.99
- Tuxedo \$90**
- Enamel Inlay Cuff Links and Stud Set:** \$24.99
- Total price \$174.97**

Right Configuration:

- Stafford® Essentials Pleated Tuxedo Pant:** \$39.99 - 49.99
- Tuxedo \$90**
- Engravable Oval Cuff Links:** \$29.99
- Total price \$169.98**

Composite retrieval problem

Given a user query Q (*central item c , a budget b*) retrieve top- k compatible satellite packages s.t. *the total cost is within budget.*

Properties of a composite item

- **Type coverage:** maximize user's exposure to as many instances of different satellite types as possible
- **Validity:** total cost of central item and compatible package is within price budget
- **Compatibility:** combine only co-purchased items
 - personalized: items co-bought by social acquaintances
- **Maximality:** build the largest *valid* package

Maximal star package (budget = \$350)



iPhone 3G 8GB \$99 + *Kroo Case \$14.95* + *Car Charger \$14.95* + *Touch Penn \$19.95* + *Portable Bose Sounddock \$149* + *iKlear Spray Kit \$24.95* = \$322.8

*forms a **valid** composite item with iPhone 3G/8GB as does any strict subset of this package.*

*forms a **maximal** package. Addition of any new item violates validity*

Top- k for maximality

- **Top- k version of a solution**
 - One inverted list per item type
 - Items sorted by price
 - For each central item, find all packages formed by compatible satellite items and pick the cheapest valid and maximal package
- **Well-know Knapsack problem for m-way joins [BDA'10 keynote]**
 - NP-hard, heuristics for two way-joins
- **What is the best strategy to avoid replicating work?**
 - Tip: maintain intermediate heaps, one for each shared satellite package
 - Tree-like structure akin to XML join processing [SIGMOD 2002]

Composite item: (social) applicability

- **Restaurant search**
 - type coverage: restaurant cuisines
 - validity: total composite cost within vacation budget
 - compatibility: places visited by friends or social acquaintances
- **Team building for problem solving [KDD 2009]**
 - type coverage: complementary expertise of team members
 - validity: size of a team
 - compatibility: team members who previously worked together

Outline

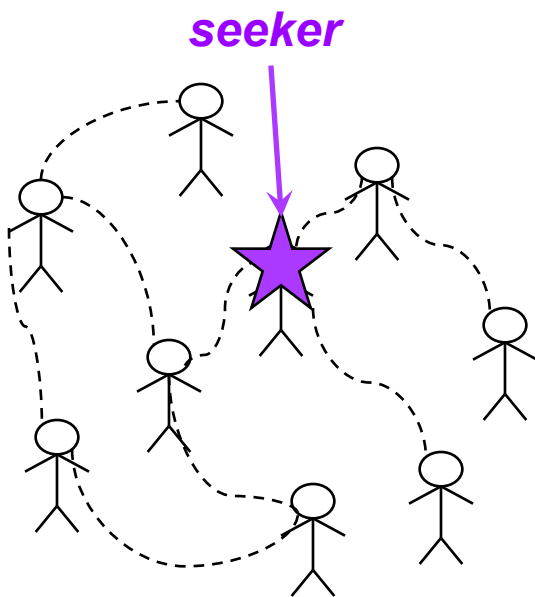
- ✓ Intro
- ✓ Composite top- k
- **Multi-top- k**
 - Problem definition
 - Overview of a solution
- **Diverse top- k**
- **Concluding thoughts**

Computing multiple top- k lists

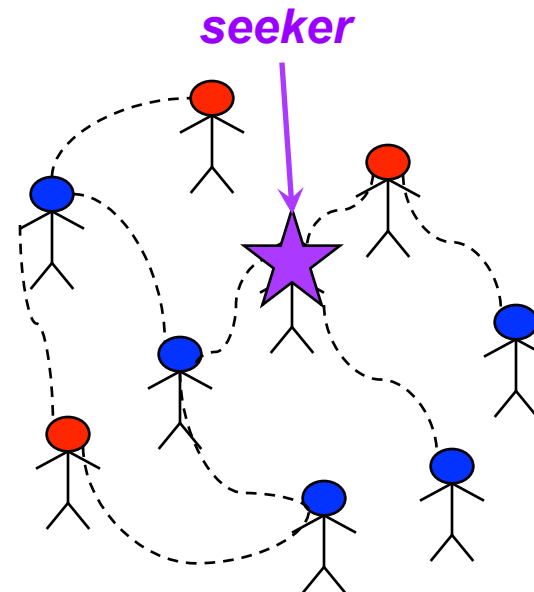
- **Recall: network-aware search**
 - Start with a *network* that gives ranking semantics, e.g., common interest based on tagged items
 - For a given user, compute a single top- k list w.r.t. that network
- **What if we wanted to compute multiple ranked lists per user, simultaneously?**
 - Define precise semantics
 - What are the right indexing structures that: (a) have reasonable space consumption and (b) support efficient processing?
 - What are the processing algorithms?

Example

Link (user u , user v)



Coloring (user u , color c)



Problem statement

- We are given a *seeker* u , and relations $Link$ (*user* u , *user* v) , $Tagged$ (*user* u , *item* i , *tag* t) and **Coloring** (*user* u , *color* c).
- A query is a set of *tags* $Q = \{t_1, t_2, \dots, t_n\}$
- For a seeker u , a tag t , an item i , and **a color** c

$score(i, u, t, c)$ = number of taggers in $Network(u)$ who:

- tagged i with c
- and are labeled with color c

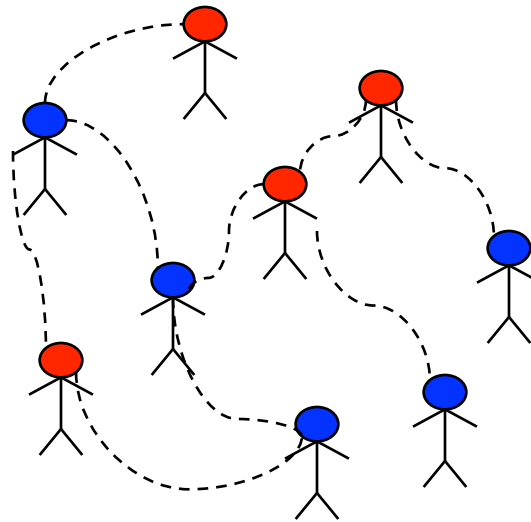
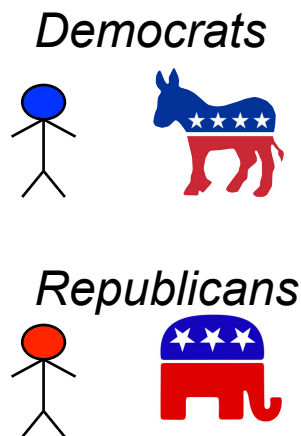
$$score(i, u, Q, c) = score(i, u, t_1, c) + \dots + score(i, u, t_n, c)$$

*Given a query Q issued by a seeker u , we wish to efficiently determine the top k items **according to each color**.*

Case 1: *disjoint networks*

- **Partition the social network**

- Each node in the network is assigned exactly one color
- Partitioning of the taggers is independent of the seeker (i.e., tagger node labels are fixed)



<i>item</i>	<i>score</i>	<i>item</i>	<i>score</i>
<i>i1</i>	99	<i>i8</i>	99
<i>i2</i>	77	<i>i9</i>	85
<i>i3</i>	58	<i>i4</i>	78
<i>i4</i>	25	<i>i7</i>	75
<i>i5</i>	23	<i>i3</i>	65
<i>i6</i>	20	<i>i6</i>	63
<i>i7</i>	15	<i>i4</i>	25
<i>i8</i>	2	<i>i1</i>	1

tag = news

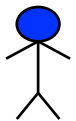
Case 1: *disjoint networks*

- **Algorithmic approach?**
 - Compute *multiple* ranked lists per seeker, each list is derived from a particular partition of the taggers
 - What type of information can be *factored out*?
 - Any other ideas?

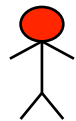
Case 2: *multi-coloring*

- **A non-partitioning assignment (networks overlap)**
 - Each node in the network is assigned 0, 1, or several colors
 - Color assignment to taggers is independent of the seeker (i.e., tagger node labels are fixed)

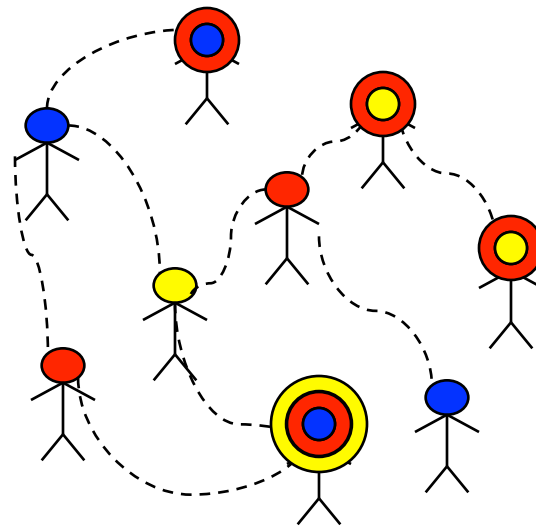
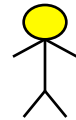
Students



Dancers



English speakers



item	score	item	score	item	score
i8	99	i1	99	i8	99
i9	85	i2	77	i9	85
i4	78	i3	58	i4	78
i7	75	i4	25	i7	75
i3	65	i5	23	i3	65
i6	63	i6	20	i6	63
i4	25	i7	15	i4	25
i1	1	i8	2	i1	1

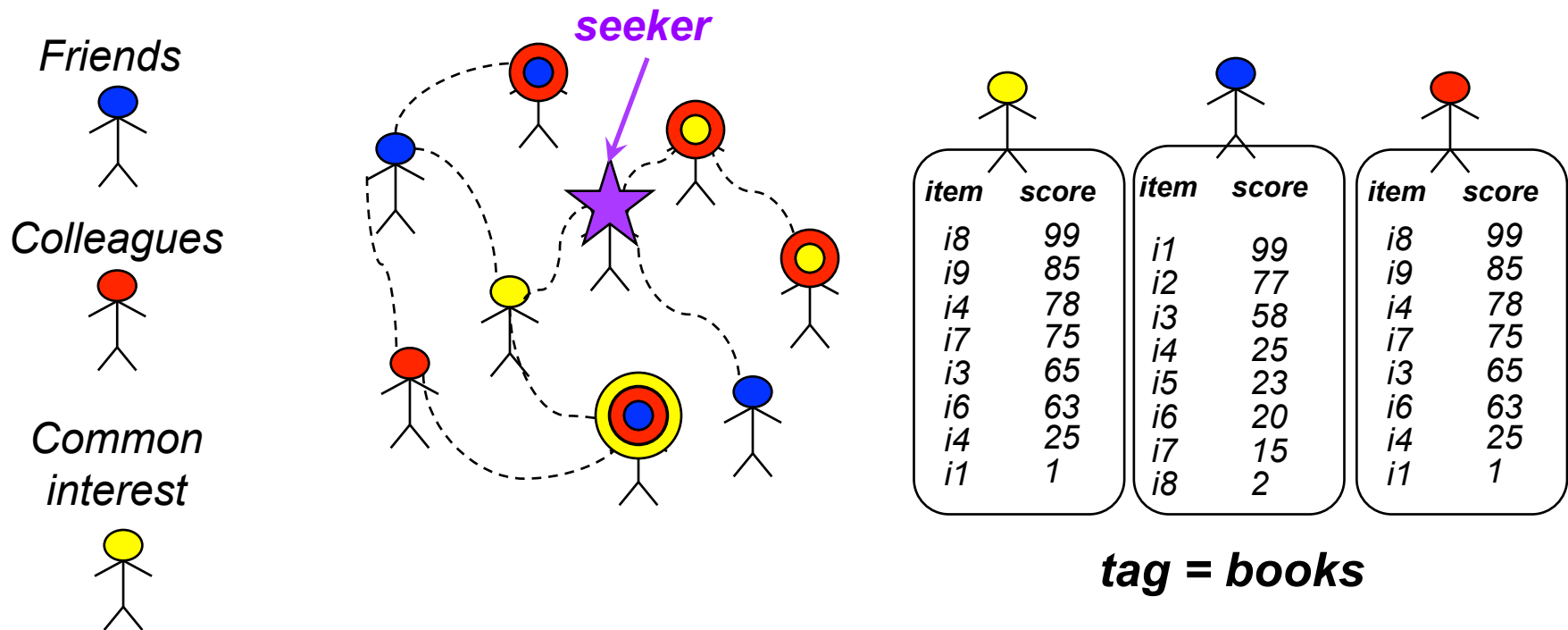
tag = clubs

Case 2: *multi-coloring*

- **Algorithmic approach?**
 - Compared to the case with disjoint networks, what are the new challenges?
 - What new information can be factored out?

Case 3: *personalized (multi-)coloring*

- **Colors are assigned to taggers depending on the seeker**
 - Each node in the network is assigned 0, 1, or several colors
 - Color assignment to taggers depends on the seeker, not fixed



Case 3: *personalized (multi-)coloring*

- **Algorithmic approach?**
 - Compared to the case with seeker-independent coloring, what are the new challenges?
 - What new information can be factored out?

Discussion

- **What are some ways of presenting multiple ranked lists to the user?**
 - Side-by-side [SIGMOD 2008 - demo]
 - Merge together into a single list, *diverse* w.r.t. source (next part)
- **What is the right type of evaluation for this application?**

Outline

- ✓ Intro
- ✓ Composite top- k
- ✓ Multi-top- k
- **Diverse top- k**
 - Problem definition
 - Overview of a solution
- **Concluding thoughts**

Diversity problem

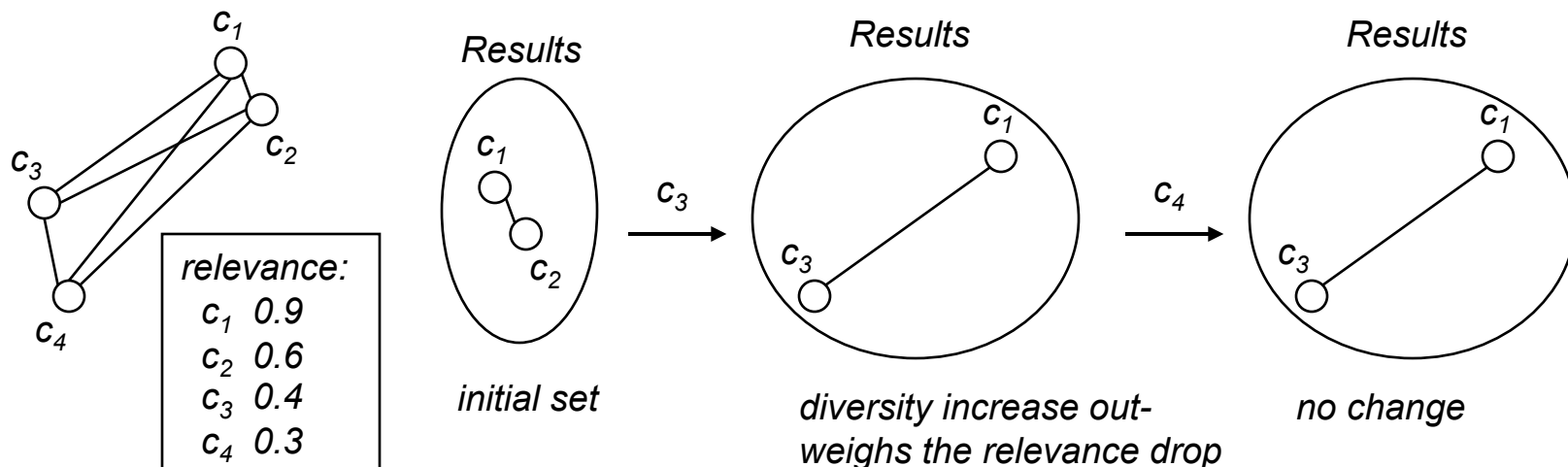
From the pool of relevant items, identify a list of items that are dissimilar to each other and maintain a high cumulative relevance, i.e., strike a good balance between relevance and diversity

Diversity challenges

- **Two kinds of diversity computations**
 - Pair-wise: a set of movies that overlap the least on genre and director
 - Set-based: a set of composite items that overlap the least on their satellite items
- **Most relevant items are not necessarily most diverse**
 - Movies most relevant to a user are all by the same 2 directors
 - Cheapest composite items all contain the same case, speaker and cleaning kit

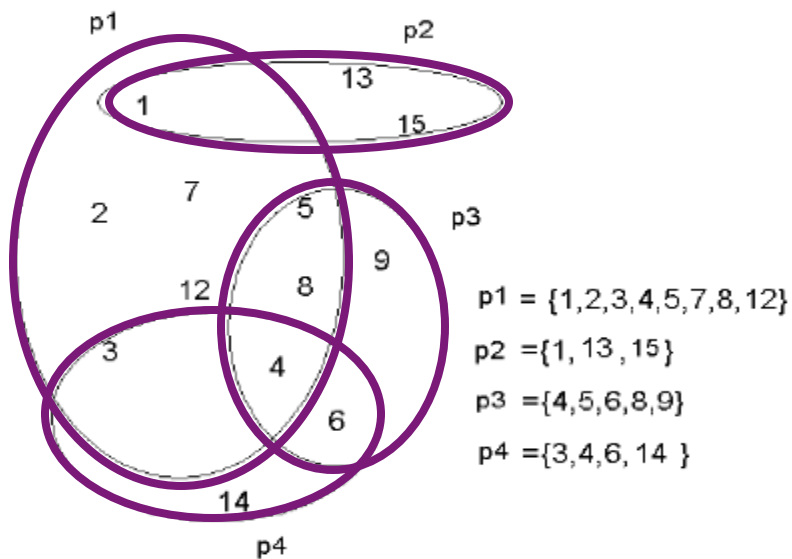
Pairwise diversity: the swap algorithm

- Sort candidate items according to their relevance
- Start by adding the K most relevant items to the result set
- Go through the rest of the candidate one by one, swap an item into the result set if the item:
 - Increases the set diversity above a certain threshold
 - Does not drop the relevance by a certain threshold
- A simple top-2 example:



Set-based diversity: k representatives

- Diversity formulated as a max- k set coverage problem and achieved by leveraging the principle of *maximizing coverage*



- p1 consists of 255 packages
- p2 consists of 7 packages
- p3 and p4 consists of 31 packages

$k = 2,$

- Best summary: {p1, p3}
- 279 packages (255 + 31 - 7)
- {p2, p3} only 38 packages

Set-based diversity



Social top-k @ Joint RuSSIR/EDBT Summer School 2011

Diversity: $k = 2$



+



+



+



+



+



+



+



Social top- k @ Joint RuSSIR/EDBT Summer School 2011

Top- k and diversity

- Find most diverse set during top- k processing
- **Tip: probe inverted lists according to distance function**
 - e.g., if distance based on attributes, partition lists on attributes and probe most diverse first, then next most diverse, etc [ICDE 2008]
- **What is the right type of evaluation for diversity?**

Outline

- ✓ Intro
- ✓ Composite top- k
- ✓ Multi-top- k
- ✓ Diverse top- k
- **Concluding thoughts**

Course summary

- **Top- k and its applications**
 - Fundamental top- k algorithms
 - Personalized search
 - Group recommendation
- **Social: user studies**
 - Group recommendation (MovieLens)
 - Travel itinerary extraction (Flickr)
- **Open problems in top- k and social**

Insights

- **Algorithmic / technical**
 - Top- k is a fundamental data processing paradigm
 - Techniques aimed at making I/O efficient, explore trade-off between processing time and space overhead
 - Applicability is wide, but not universal: ranking functions must be monotone!
- **Usability**
 - All user-facing applications, e.g., social search and recommendation, are aimed at user satisfaction
 - Semantics must be realistic
 - Validation with real datasets and user studies is essential!
- **Novelty**
 - The Social Web is young, it will become what we, as users, researchers and developers make of it!
 - Many unexplored technical opportunities over-all, and also in top- k

References and further reading

Two keynotes at BDA 2010

1. *Top-k knapsack joins*. Witold Litwin, Thomas Schwarz. <http://www.irit.fr/BDA2010/cours/LitwinBDA10.pdf>
2. *Composite Retrieval of Stars and Chains*. Sihem Amer-Yahia. <http://www.irit.fr/BDA2010/cours/AmerYahiaBDA10.pdf>
3. *Holistic twig joins: optimal XML pattern matching*.
Nicolas Bruno, Nick Koudas, Divesh Srivastava. SIGMOD 2002.
4. *Battling Predictability and Overconcentration in Recommender Systems*.
Sihem Amer-Yahia, Laks Lakshmanan, Sergei Vassilvitskii, Cong Yu. DEBU 2009.
5. *From del.icio.us to x.qui.site: Recommendations in Social Tagging Sites*. Sihem Amer-Yahia, Alban Galland, Julia Stoyanovich, Cong Yu. SIGMOD 2008 (demo).
6. *Efficient Computation of Diverse Query Results*. Erik Vee. Utkarsh Srivastava, Jayavel Shanmugasundaram, Prashant Bhat, Sihem Amer-Yahia. ICDE 2008.
7. *Finding a team of experts in social networks*.
Theodoros Lappas and Kun Liu and Evimaria Terzi. KDD 2009.

Questions?

