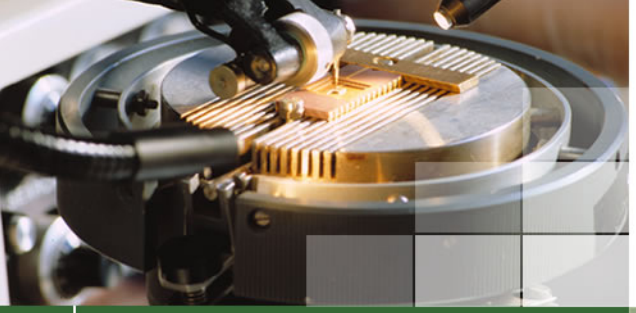


POLITECNICO DI MILANO

Dipartimento di
Elettronica e Informazione

Ranking and queries: as good as it gets

Davide Martinenghi

Lugano, July 5, 2012

My main research areas

- Data Integrity Checking
  - Integrity constraints are properties that represent the legal states of a database
  - How to best preserve full satisfaction of constraints in the face of updates? (incremental integrity maintenance)
  - What to do when we update a database that already violates some constraints? (inconsistency-tolerant integrity checking)
- Query Answering over the Web
  - How to answer queries over data behind forms (Deep Web)? (query answering under access limitations)
  - Lots of distinctive (but often implicit) aspects of data on the Web
    - recency
    - incompleteness of information
    - different levels of granularity in the data
    - uncertainty
    - provenance
- Ranking queries (this talk)

Outline

- Ranking queries
- Rank aggregation
  - Based on position
  - Aggregation functions
- Ranking in the real world
  - Joins
  - Proximity
  - Uncertainty
  - Diversity
- Future directions

Ranking queries

- Main idea: focus on the best query answers according to some criterion, without computing the full result
  - A.k.a. "top-k" queries
- Main applications:
  - Combination of user preferences expressed according to various criteria
    - Example: ranking restaurants by combining criteria about culinary preference, driving distance, stars, …
  - Nearest neighbor problem (e.g., similarity search)
    - Given a database *D* of *n* points in some metric space, and a query *q* in the same space, find the point (or the *k* points) in *D* closest to *q*
  - Search computing
    - "Where can I attend an interesting conference in my field close to a sunny beach?"
  - …

Ranking queries: example


```
SELECT h.neighborhood, h.hid, r.rid
FROM HotelsNY h, RestaurantsNY r
WHERE h.neighborhood = r.neighborhood
RANK BY 0.4/h.price + 0.4*r.rating + 0.2*r.hasMusic
LIMIT 5
```

Full Join Results

Neighborhood	Hid	Rid
West Village	H89	R585
Midtown East	H248	R197
Chelsea	H427	R572
Midtown East	H248	R346
Midtown East	H597	R197
Hell's Kitchen	H662	R223
Midtown West	H141	R276
Upper East Side	H978	R137
Harlem	H355	R49
Tribeca	H381	R938
. . .	. . .	. . .

Rank Join Results

Neighborhood	Hid	Rid
East Village	H346	R738
Gramercy	H872	R822
Midtown West	H141	R276
Hell's Kitchen	H662	R498
Upper West Side	H51	R394

Rank aggregation (your problem)

[Fagin, PODS 1996]

- Rank aggregation is the problem of combining several ranked lists of objects in a robust way to produce a single consensus ranking of the objects

Candidate	Candidate	Candidate	Candidate	Candidate
1	2	4	5	3
2	4	2	1	5
3	5	5	3	1
4	1	3	4	2
5	3	1	2	4
Judge 1	Judge 2	Judge 3	Judge 4	Judge 5

- What is the overall ranking?
- Who is the best candidate?

Rank aggregation and scores

- Metric approaches are preferred over axiomatic approaches (Arrow's impossibility theorem)
- When scores are opaque, the goal is to find a new ranking R whose total distance to the initial rankings R_1, …, R_n is minimized
  - For several metrics, NP-hard to solve exactly
    - E.g., the **Kendall tau distance** $K(R_1, R_2)$, defined as the number of exchanges in a bubble sort to convert R_1 to R_n
  - May admit efficient approximations (e.g., median ranking)
- When scores are visible, the consensus ranking is determined by an aggregation function

Rank aggregation – example with scores

- Aggregation function:

$$\text{Score(cand)} = 0.30\, s_1 + 0.25\, s_2 + 0.20\, s_3 + 0.15\, s_4 + 0.10\, s_5$$

Cand	s_1
1	.9
2	.7
3	.5
4	.3
5	.1

Judge 1

Cand	s_2
2	.65
1	.6
5	.55
4	.5
3	.45

Judge 2

Cand	s_3
4	.99
2	.97
5	.95
3	.93
1	.91

Judge 3

Cand	s_4
5	.6
1	.5
3	.4
4	.3
2	.2

Judge 4

Cand	s_5
3	.8
1	.7
5	.65
2	.63
4	.62

Judge 5

- What is the overall ranking?
- Who is the best candidate?

Reverse top-k queries (my problem)

[Vlachou et al., ICDE 2010]

- Aggregation function:

  Score(cand) = $w_{SIGMOD}\ s_1 + w_{VLDB}\ s_2 + w_{ICDE}\ s_3 + w_{TODS}\ s_4 + w_{TKDE}\ s_5$

Full papers in the top database venues in the last 5 years

Cand	s_1
1	2
4	2
2	0
3	0
5	0

SIGMOD

Cand	s_2
4	4
1	2
2	0
3	0
5	0

VLDB

Cand	s_3
1	1
2	1
4	1
3	0
5	0

ICDE

Cand	s_4
1	1
5	0
2	0
3	0
4	0

TODS

Cand	s_5
1	2
4	1
2	0
3	0
5	0

TKDE

- What weights should I convince you to use so that I become the best candidate?
  - (point of view of the seller/product manufacturer)

Rank aggregation in data-centric contexts

- Traditionally, two ways of accessing data:
  - Sorted access: access, one by one, the next element (together with its score) in a ranked list, starting from top
  - Random access: given an element (id), retrieve its score (position in the ranked list or other associated value)
- Minimizing the accesses when determining the top k items
  - A cost is incurred for each item read from a ranking
  - Can I improve on the current best aggregate score if I read more items?
  - Thresholds are used to ensure that no further item needs to be read

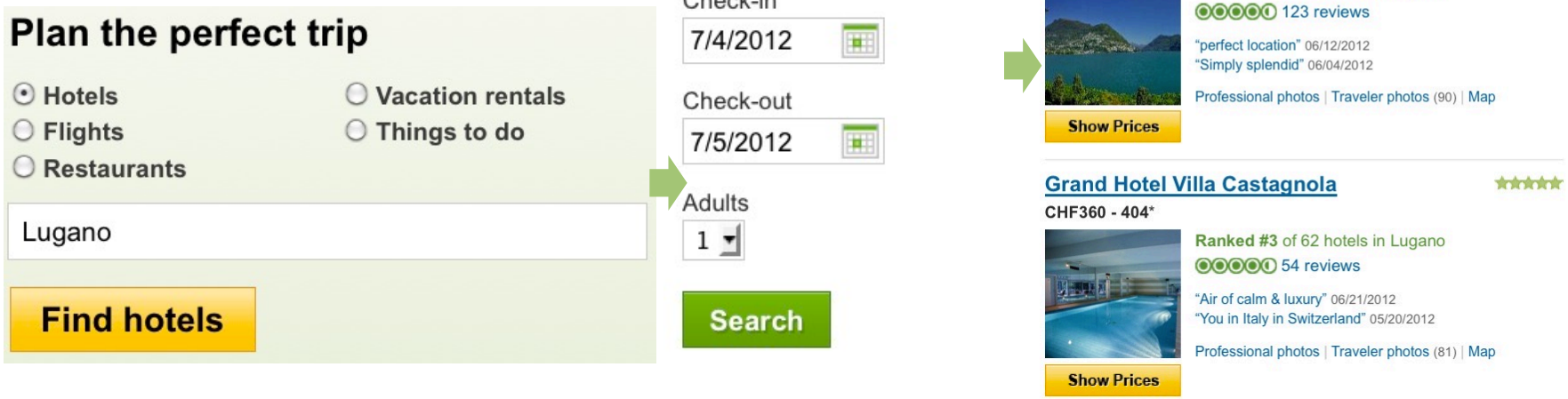
Ranking in the real world

[Calì & Martinenghi, ICDE 2008] [Martinenghi & Tagliasacchi, TKDE 201X]

- Almost relational model, with a lot of "quirks"
  - Web interfaces with input and output fields (access patterns)
  - Results are typically ranked

tripAdvisor(City^{i}, InDate^{i}, OutDate^{i}, Persons^{i}, Name^{o}, Popularity^{o,ranked})

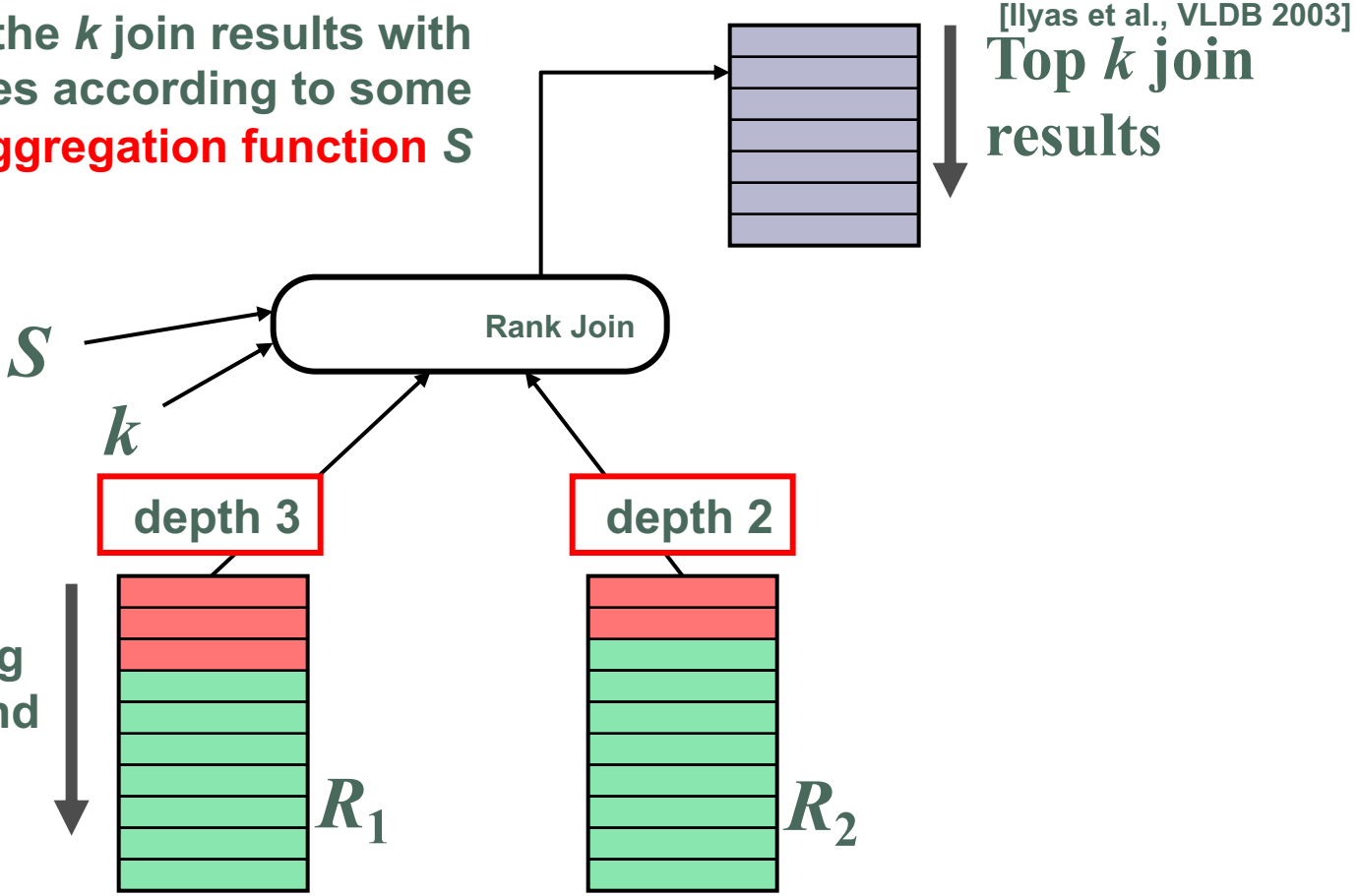
  - Many other needs: joins, dirty data, deduplication, diversification, uncertainty, incompleteness, recency, paging, access costs…



The Rank Join Problem

Computing the *k* join results with highest scores according to some aggregation function *S*

[Ilyas et al., VLDB 2003]



- Total depth (aka sumDepths) is the primary cost metric
- Early termination: no algorithm can be optimal
  - But an algorithm can be instance optimal, i.e., the best possible algorithm (to within a constant factor) on every input instance

Proximity Rank Join: example

[Martinenghi & Tagliasacchi, VLDB 2010]
[Martinenghi & Tagliasacchi, TODS 2012]

- A smartphone user wants to organize the evening by finding:
  - a restaurant, a movie theater and a hotel that are
    - nearby
    - close to each other
    - recommended in terms of price, user rating, and number of stars

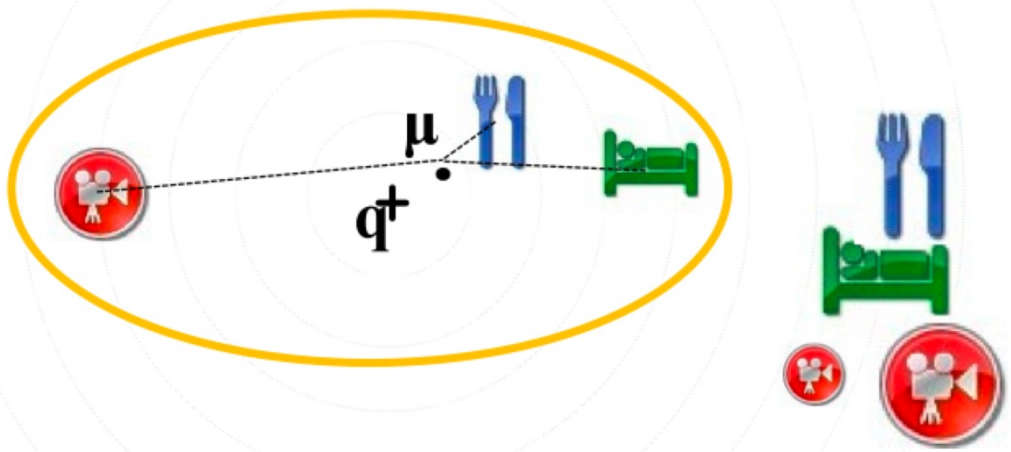
Proximity rank join problems

- Looking for combinations of heterogeneous objects
- Each object is equipped with
  - A score
  - A real-valued feature vector

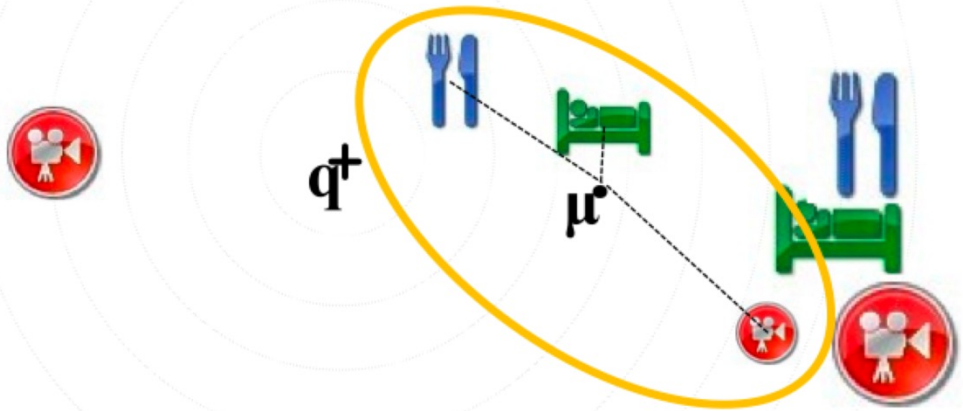
Hotel	Category	Location
Villa D'Este	5	[45.62 N, 9.32 E]
Metropole Suisse	4	[45.65 N, 9.33 E]
Palace Hotel	4	[45.64 N, 9.31 E]

- The aggregation function assigns a score to a combination based on
  - The individual scores
  - The proximity to the query vector
  - Their mutual proximity
- Objects can also be retrieved by distance from the query

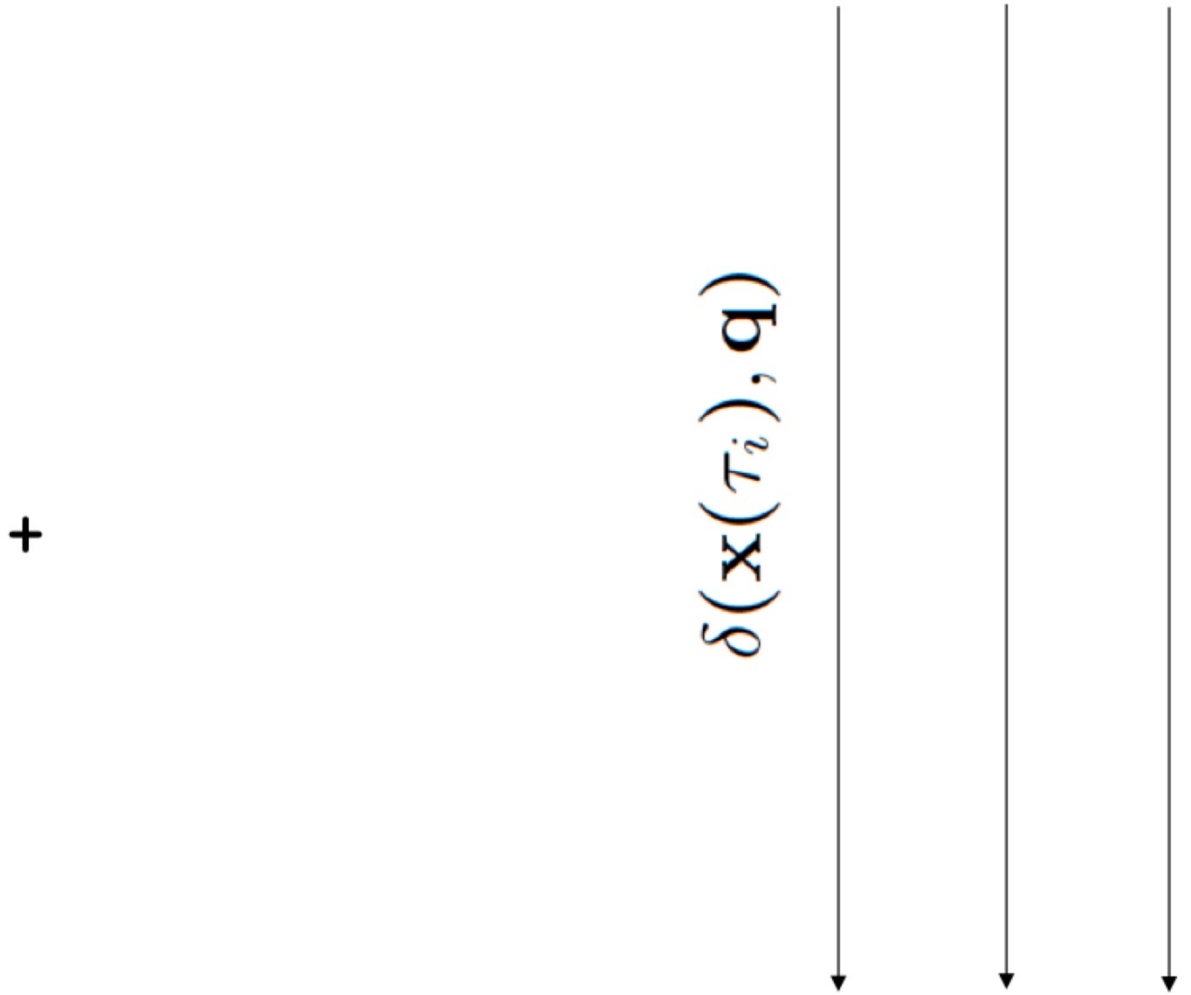
Proximity Rank Join



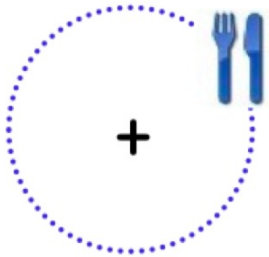
Proximity Rank Join



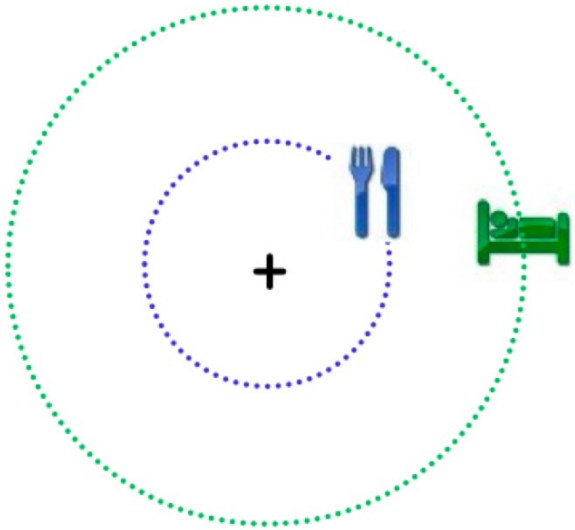
Distance-based access

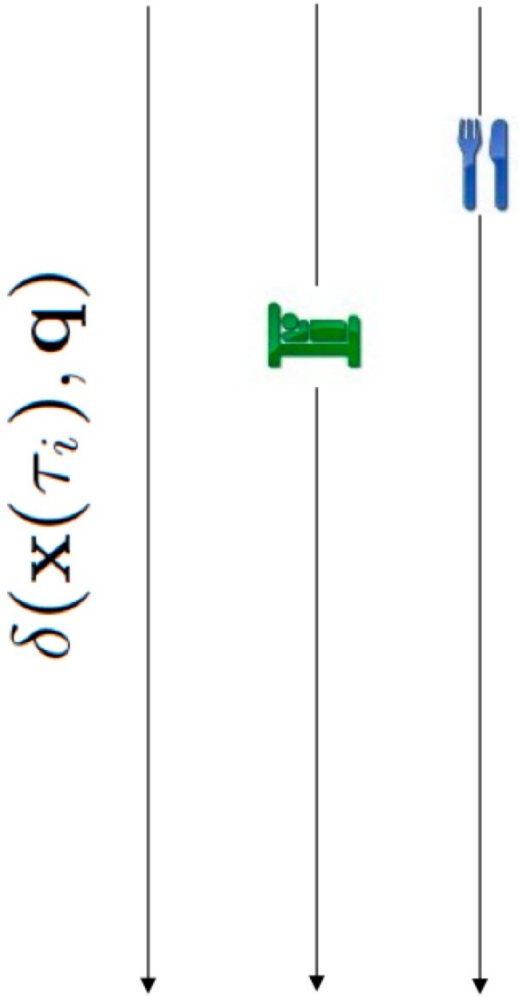


Distance-based access

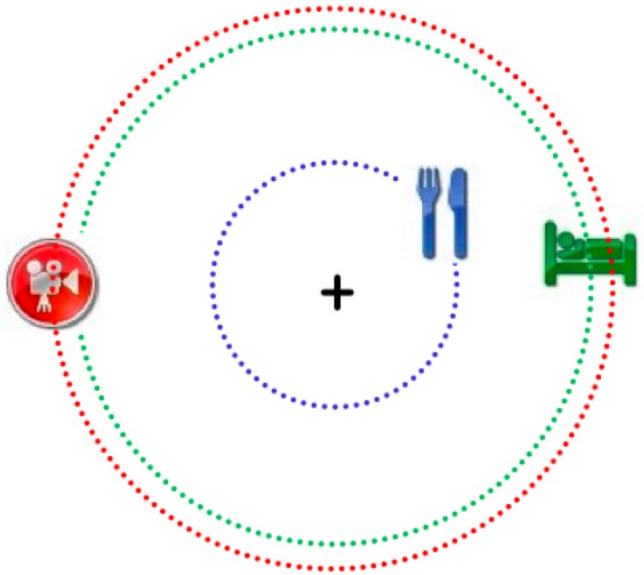


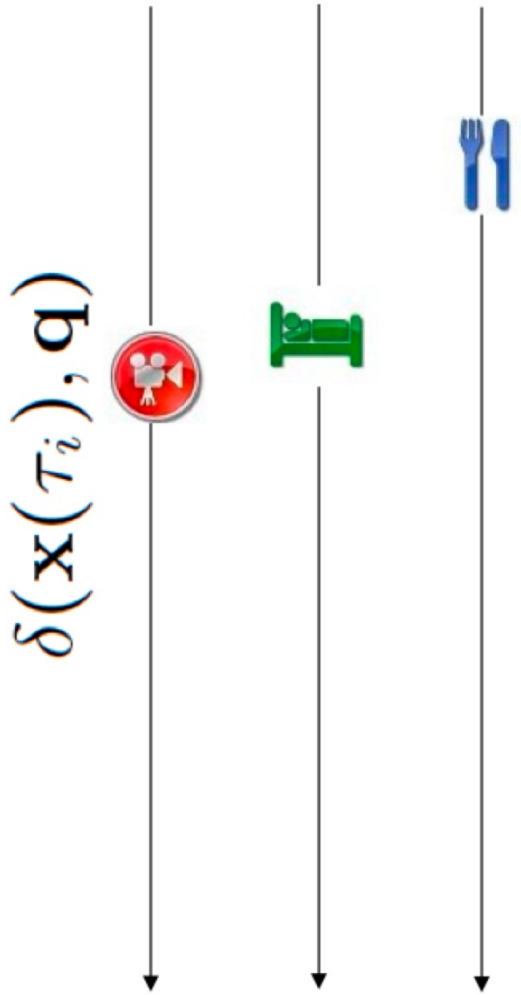
Distance-based access



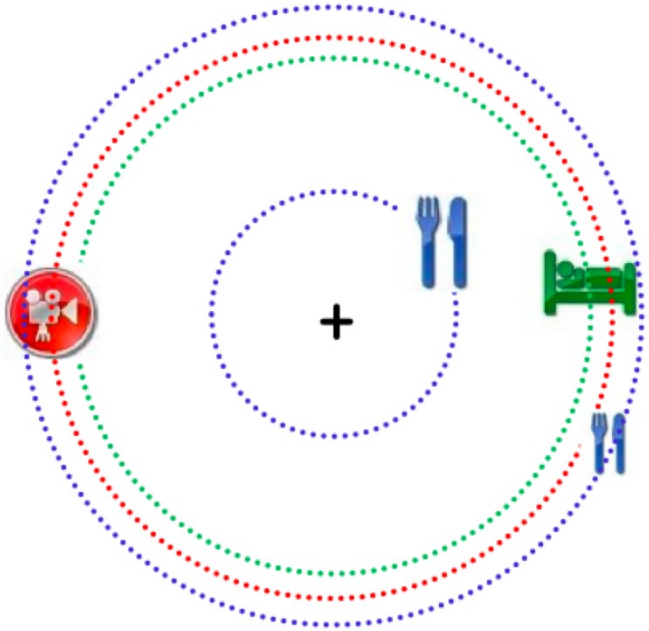


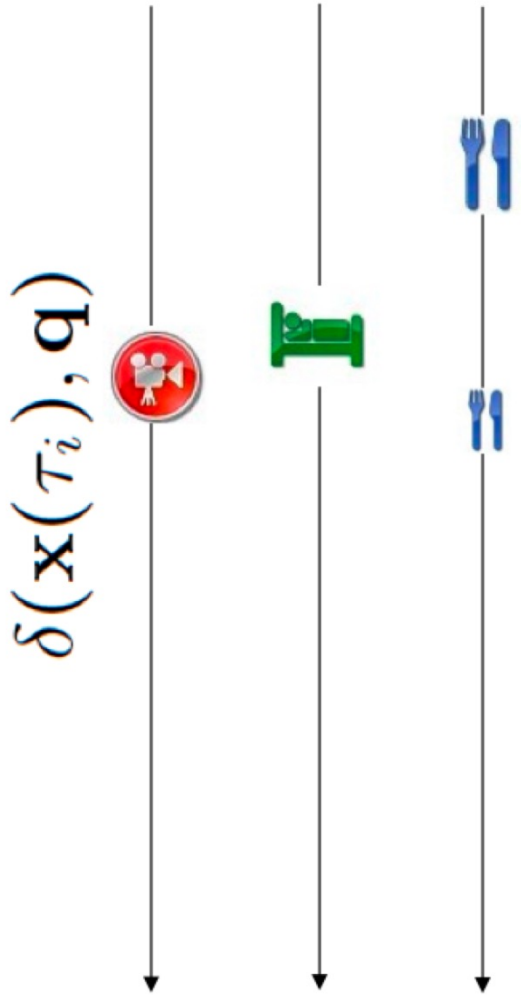
Distance-based access



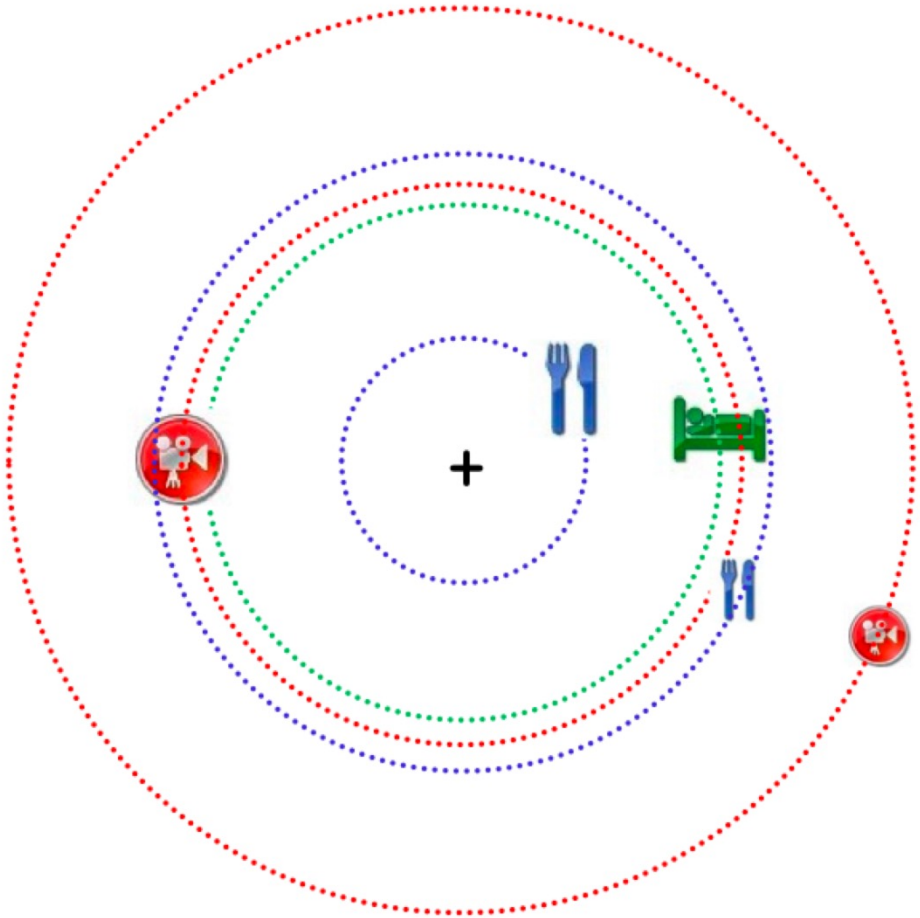


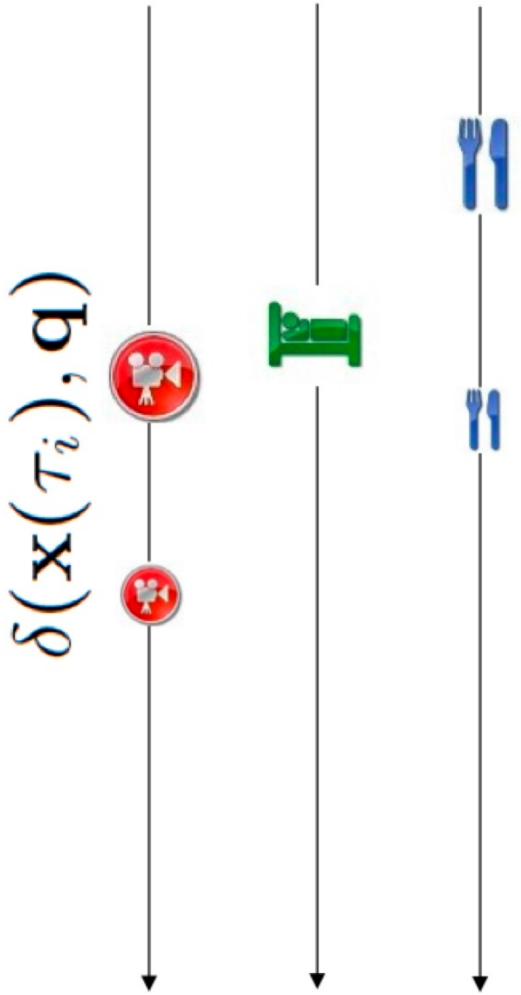
Distance-based access



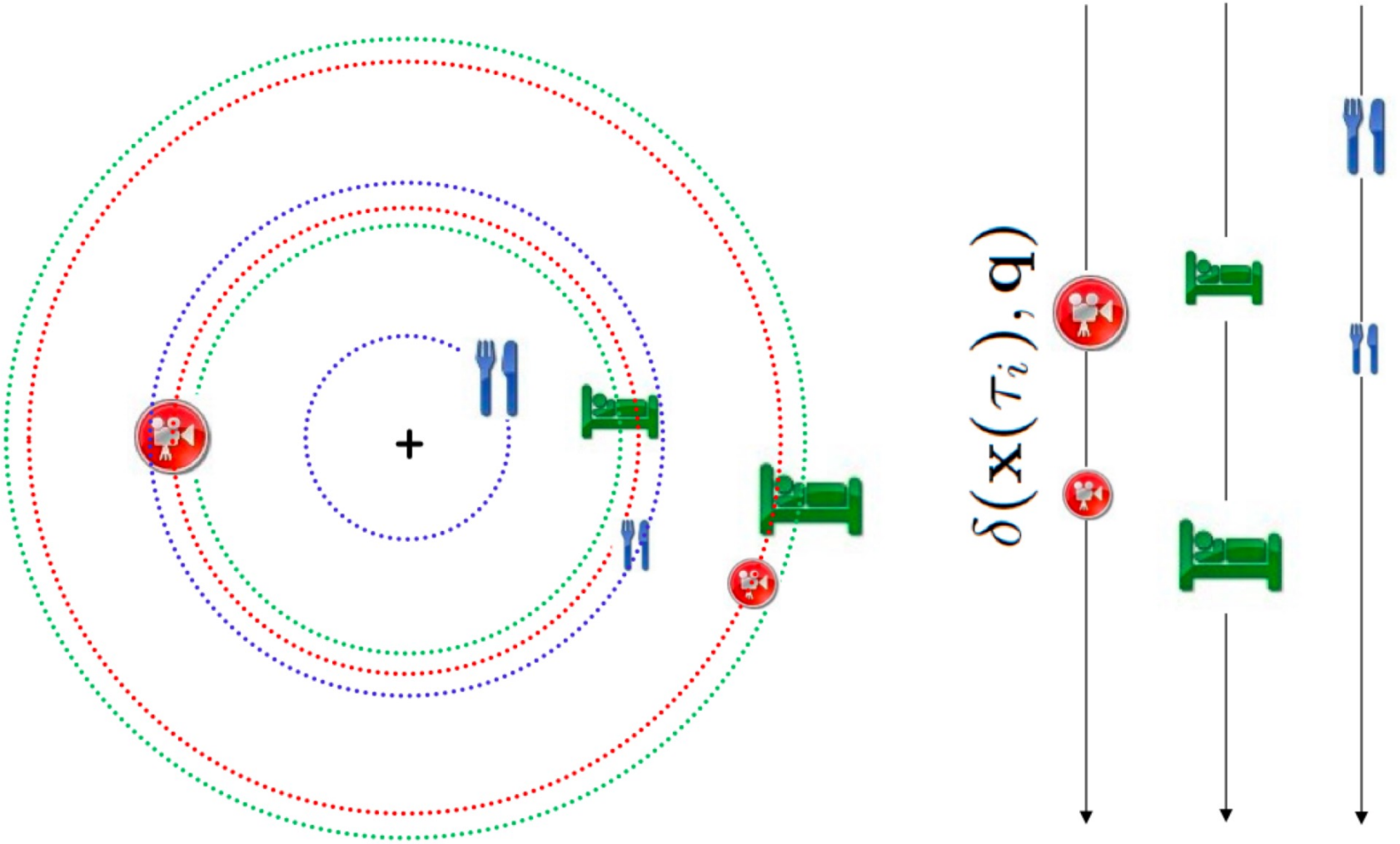


Distance-based access





Distance-based access



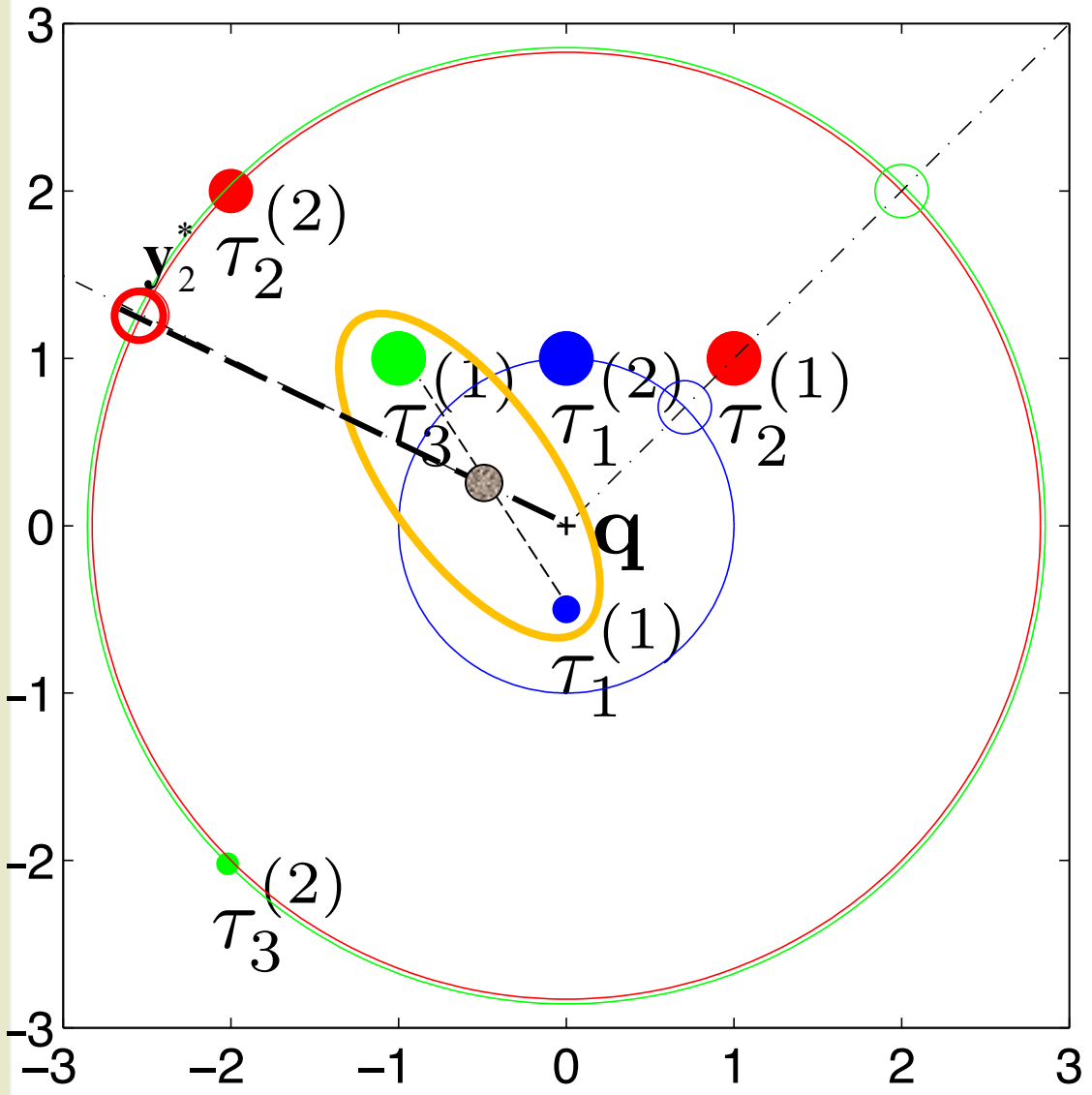
Proximity rank join problems

- Broad applicability
  - Information retrieval
    - E.g. finding similar documents in different collections given a set of keywords
    -
  - Multimedia databases
    - E.g. requesting similar images from different repositories given a sample image
  - Bioinformatics
    - E.g. discovering orthologous genes from different organisms given a target annotation profile

Bounding scheme

- Stopping criterion based on a bounding scheme:
  - What is the largest aggregate score of a possible combination formed with at least one unseen tuple?
  - We stop when we have k combinations whose score exceeds the bound
- Tight bound (an actually achievable bound)
  - Using tight bounds guarantees instance optimality
  - Can be computed efficiently when using Euclidean distance

Tight bound for Euclidean distance



M	$\tau \in PC(M)$	$t(\tau)$
$\varnothing$	$\langle\rangle$	-19.2
$\{1\}$	$\tau_1^{(1)}$	-20.6
	$\tau_1^{(2)}$	-19.2
$\{2\}$	$\tau_2^{(1)}$	-12.8
	$\tau_2^{(2)}$	-19.4
$\{3\}$	$\tau_3^{(1)}$	-12.8
	$\tau_3^{(2)}$	-20.1
$\{1,2\}$	$\tau_1^{(1)} \times \tau_2^{(1)}$	-16.0
	$\tau_1^{(1)} \times \tau_2^{(2)}$	-24.0
	$\tau_1^{(2)} \times \tau_2^{(1)}$	-13.5
	$\tau_1^{(2)} \times \tau_2^{(2)}$	-20.4
$\{1,3\}$	$\tau_1^{(1)} \times \tau_3^{(1)}$	-16.0
	$\tau_1^{(1)} \times \tau_3^{(2)}$	-22.0
	$\tau_1^{(2)} \times \tau_3^{(1)}$	-13.5
	$\tau_1^{(2)} \times \tau_3^{(2)}$	-26.4
$\{2,3\}$	$\tau_2^{(1)} \times \tau_3^{(1)}$	-7.0
	$\tau_2^{(1)} \times \tau_3^{(2)}$	-21.0
	$\tau_2^{(2)} \times \tau_3^{(1)}$	-13.1
	$\tau_2^{(2)} \times \tau_3^{(2)}$	-26.8

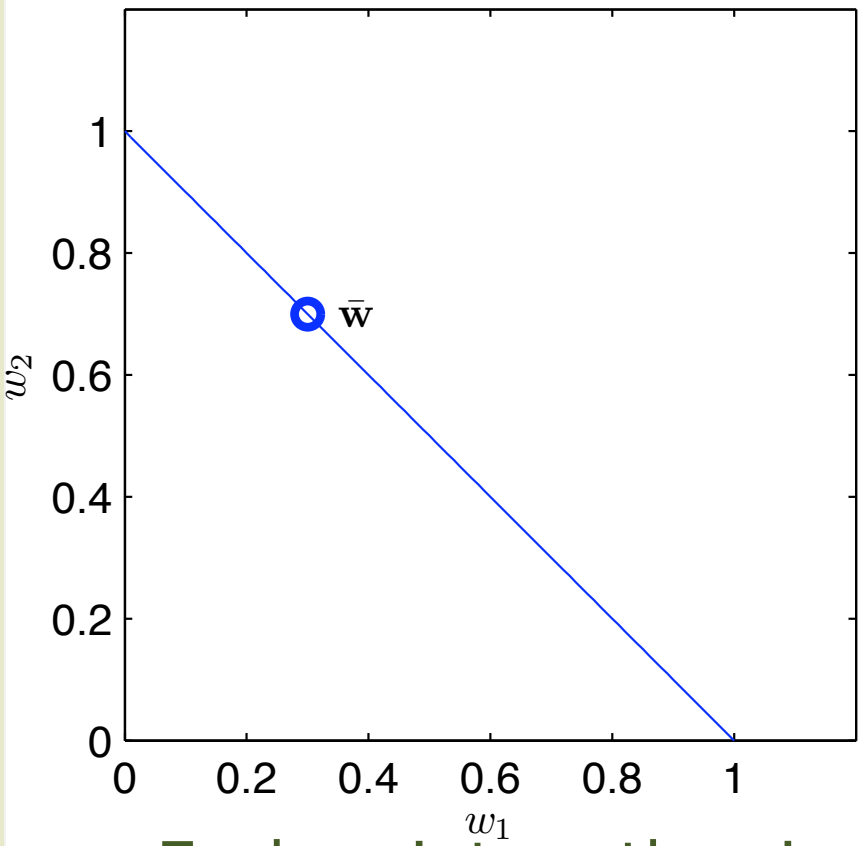
Uncertainty in rank join

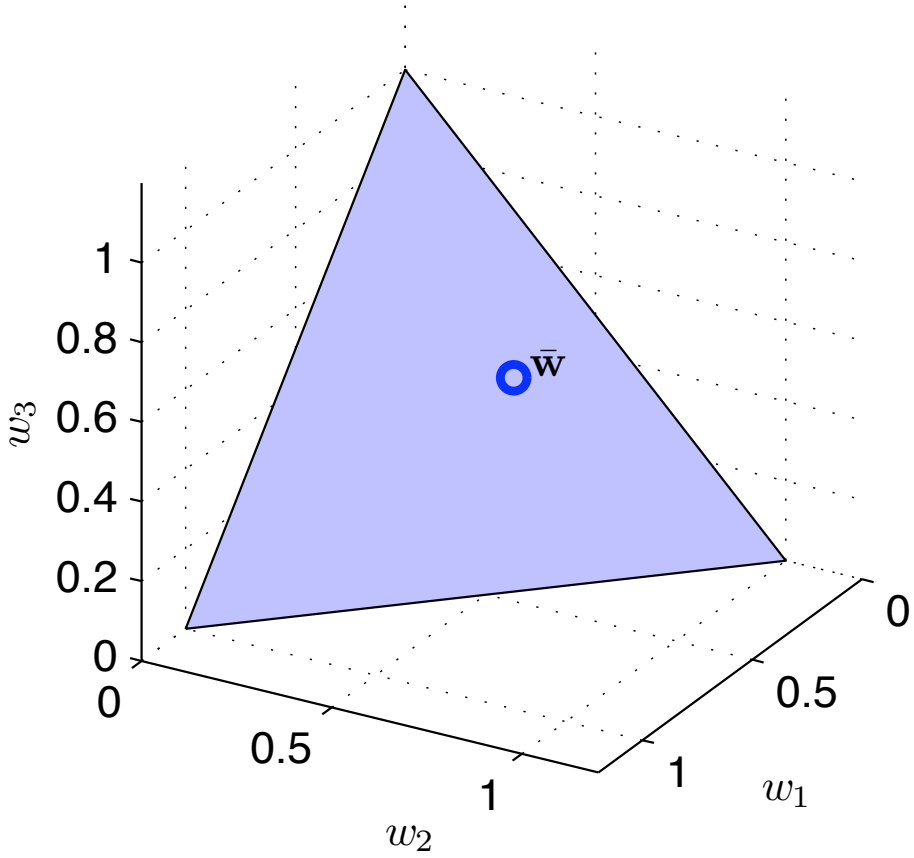
[Soliman, Ilyas, Martinenghi, Tagliasacchi, SIGMOD 2011]

- Users are often unable to precisely specify the scoring function
- Using trial-and-error or machine learning may be tedious and time consuming
- Assumptions:
  - Linear scoring function:

    $S = w_1 s_1 + w_2 s_2 + \ldots + w_n s_n$
  - User-defined weights $w_1, w_2, \ldots, w_n$ are:
    - uncertain, and, w.l.o.g.,
    - normalized to sum up to 1
- *[Part of a current FET proposal, second round]*

Representing scoring functions on the simplex





- Each point on the simplex represents a possible scoring function
- We assume that p(**w**) is uniform over the simplex

Uncertain scoring

- Uncertainty induces a probability distribution on a set of possible rankings
- Each ordering occurs with a probability

$$p(\boldsymbol{\lambda}_N) = \int_{\mathbf{w} \in \Delta^{d-1}, \mathcal{O} \overset{\mathbf{w}}{\rightsquigarrow} \boldsymbol{\lambda}_N} p(\mathbf{w}) d\mathbf{w}$$

(weights in the simplex inducing that ranking)

- When N is large, we usually focus on a prefix of length K<N of an ranking

Example (our problem, again)

- Top-k query

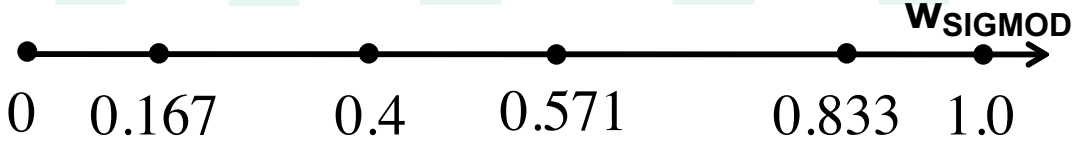
SELECT candidate, s_{SIGMOD}, s_{TODS}
FROM SIGMOD, TODS
RANK BY w_{SIGMOD} s_{SIGMOD} + w_{TODS} s_{TODS}
LIMIT 1

- Results and possible rankings

Candidate	S_{SIGMOD}	S_{TODS}
1	7	5
2	2	6
3	4	7
4	5	2

(w_{SIGMOD} + w_{TODS} = 1)

A	B	C	D	E
3	3	1	1	1
2	1	3	3	4
1	2	2	4	3
4	4	4	2	2



Representative orderings

- Finding a representative ranking:
  - Most Probable Ordering:

$$\boldsymbol{\lambda}^*_{MPO} = arg. \max_{\boldsymbol{\lambda} \in \Lambda_K} p(\boldsymbol{\lambda})$$

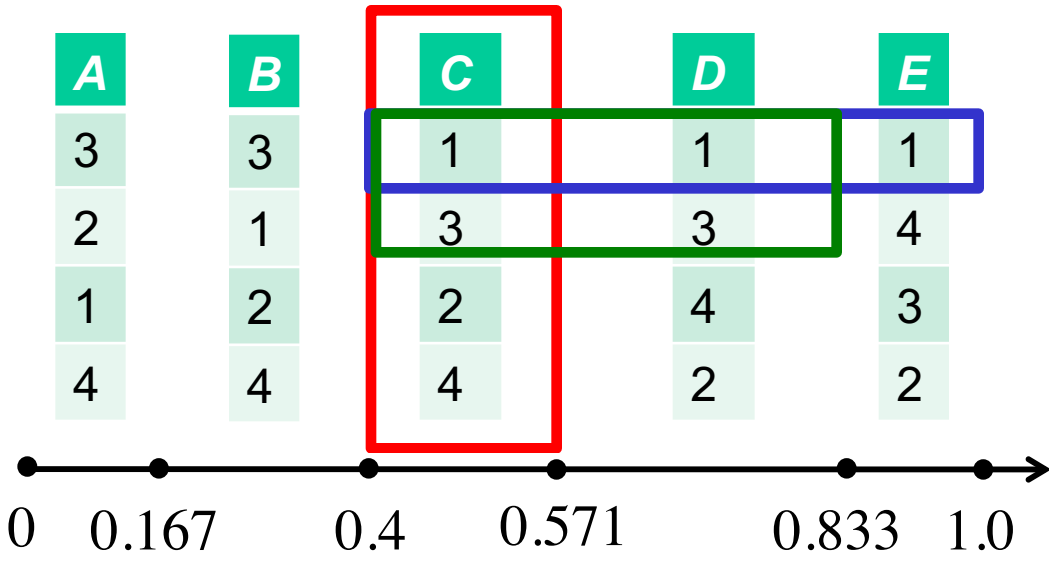
  - Optimal Rank Aggregation:
    - Ranking with the minimum average distance to all other rankings
  - Common distances between rankings:
    - Kendall tau: number of pairwise disagreements in the relative order
    - Spearman's footrule: sum of distances between the ranks of the same item in the two rankings

Problem	$d = 2$	$d = 3$	$d > 3$
MPO (average case)	$O(N(logN)^{K+1})$	$O(N(logN)^{2K+1})$	$O(N^{\lfloor d/2 \rfloor + 1}(logN)^{(d-1)K})$ [§]
MPO (worst case)	$O(N^2 logN)$	$O(N^4)$	$O(N^{2^{d-1}})$ [§]
ORA (Kendall tau)	$O(NlogN)$	NP-Hard	NP-Hard
ORA (Footrule)	$O(N^{2.5})$	$O(N^4)$	$O(N^{2^{d-1}})$ [§]

Example of MPO and ORA

- For K=1, the MPO is <1>
- For K=2, the MPO is <1,3>
- ORA is *C* both for Kendall tau and footrule

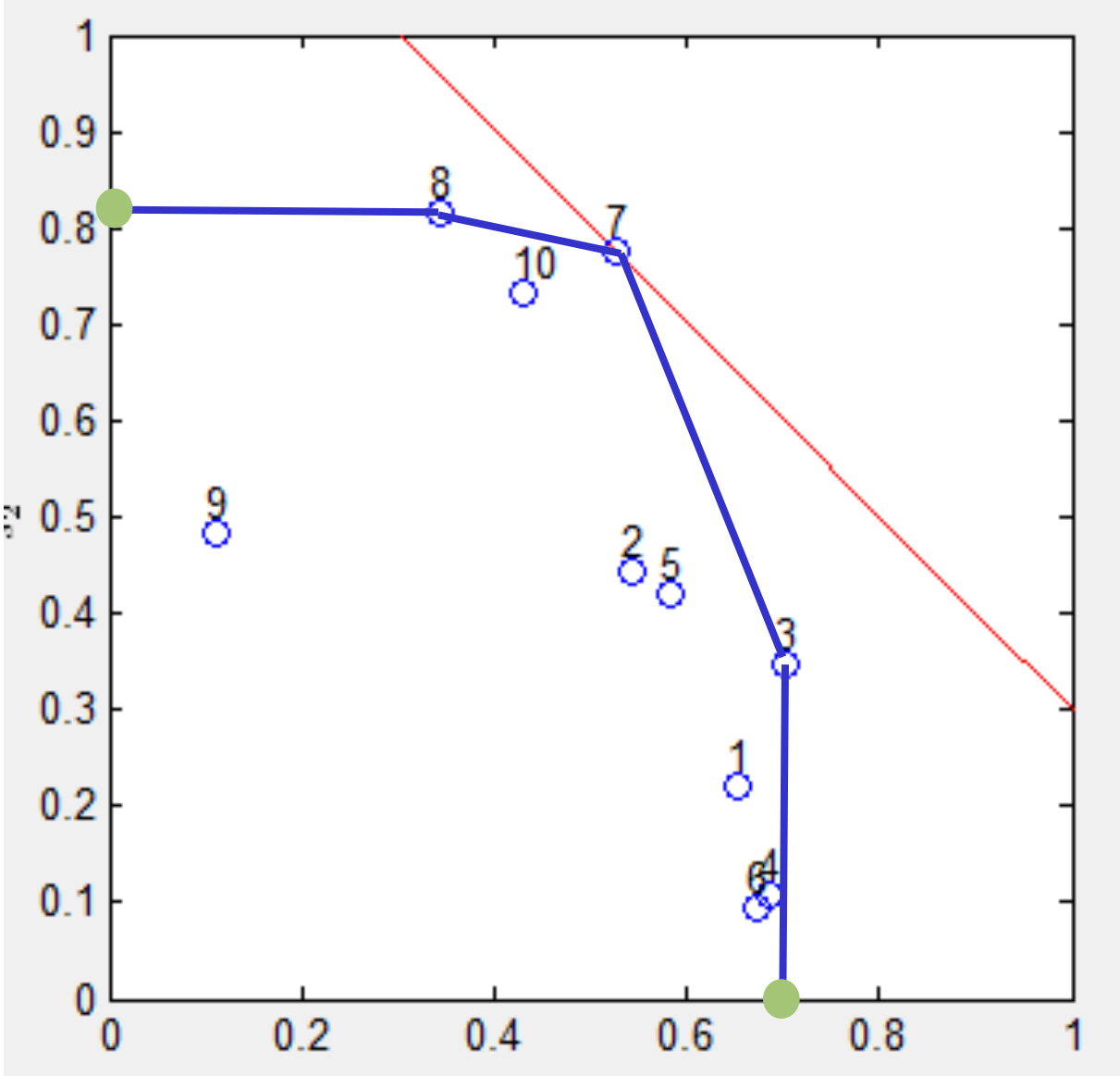
Candidate	S_{SIGMOD}	S_{VLDB}
1	7	5
2	2	6
3	4	7
4	5	2



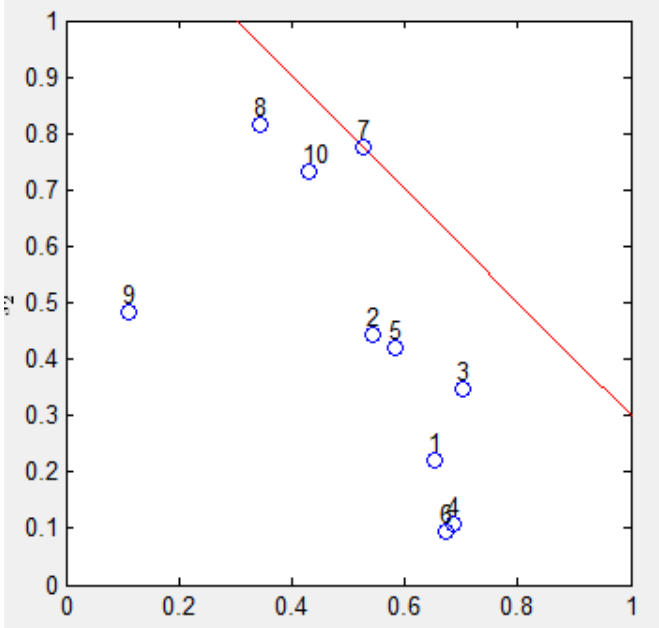
Computing representative rankings

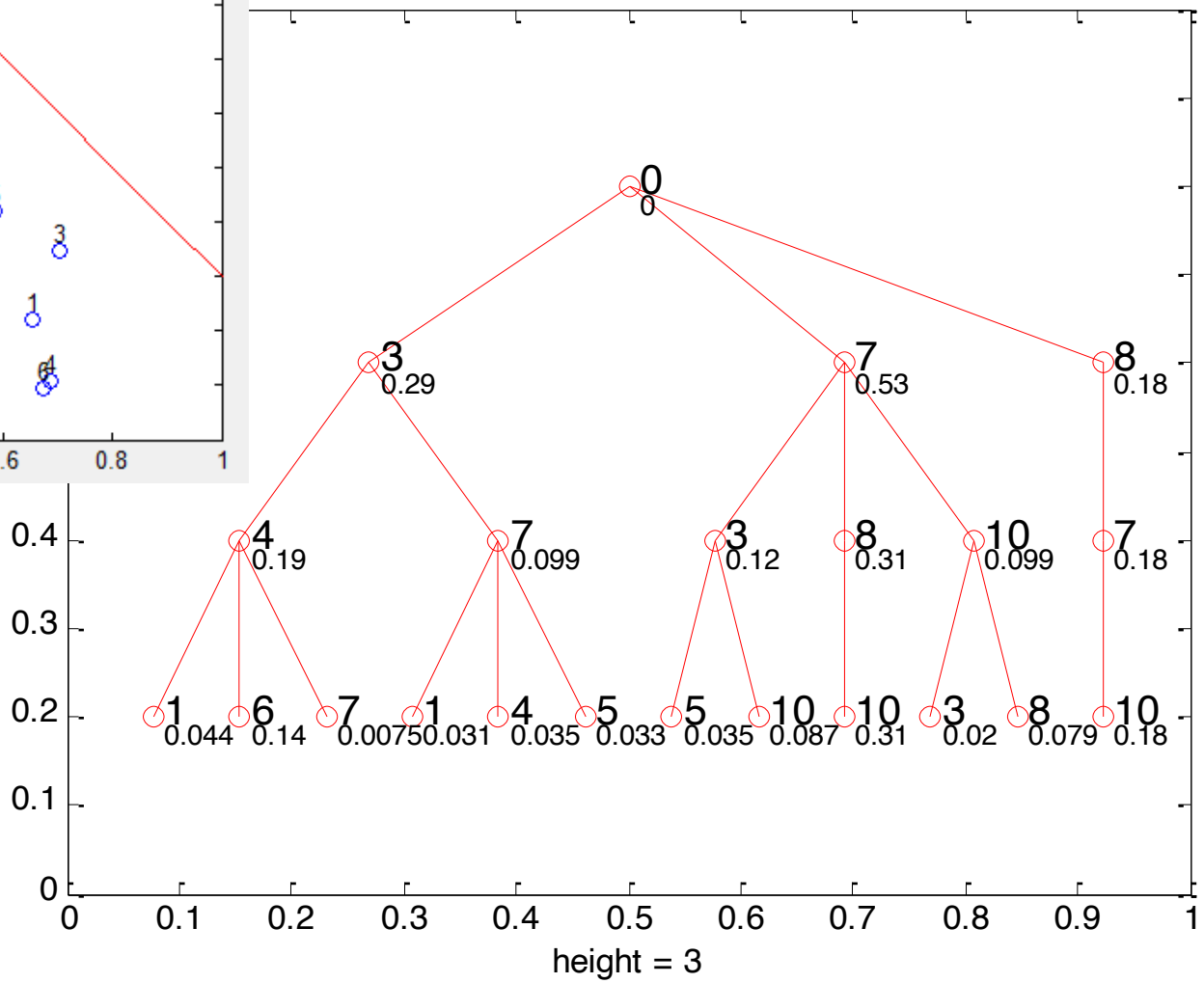
- Naïve approach:
  1. Enumerate possible weight vectors
  2. Find the distinct rankings induced by these vectors
  3. Pick the required representative ranking
- This is:
  - Highly inefficient
  - Inaccurate, since it requires discretizing the weights space
- An incremental approach: tree-based representation that is incrementally constructed by extending prefixes of rankings
  - Appropriate for MPO

Results in 2D and uncertain scoring function



Incremental construction of the possible rankings





Conclusions and future work

- Rank aggregation: merging rankings into a consensus list
- Rank join
  - Extension to heterogeneous (joinable) relations
  - Requires sorted access to data (or even random access)
    [TKDE 201X]
  - Efficiency measured as total depth (aiming at instance optimality)
- Extensions
  - In proximity r. j. objects are in a vector space affecting the score
    [VLDB 2010] [TODS 2012]
  - With uncertain scoring, we look for representative rankings
    [SIGMOD 2011] [FET proposal, 2nd round]
  - Diversification of results (not discussed in this talk)
    [SIGMOD 2012]
- Future work
  - Use human computing to reduce uncertainty: what is the most promising question to ask a human so as to crystallize the MPO?

As good as it gets

[Braga, Ceri, Daniel, Martinenghi, VLDB 2008]

- "Where can I attend an interesting conference in my field close to a sunny beach?"

Query input

Conference topic: Database

Min. temperature: 28

search

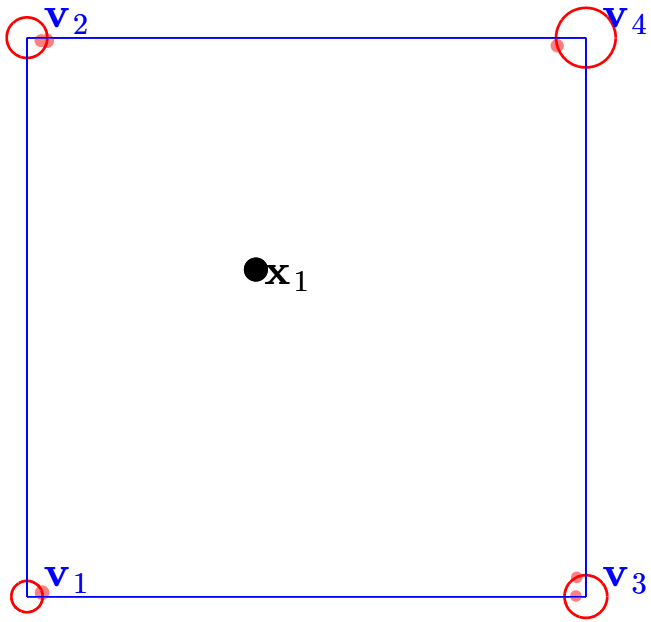
Search Results

Conf	City	FPrice	Start	End	Hotel	HPrice	Star
MSVVEIS 2008	Barcelona	234.00	12/06/2008	13/06/2008	Hotel Silken Diagonal Bar	81.00	10.1
MSVVEIS 2008	Barcelona	234.00	12/06/2008	13/06/2008	Moderno	88.00	10.1
MSVVEIS 2008	Barcelona	234.00	12/06/2008	13/06/2008	Hotel 1898	89.00	10.1
MSVVEIS 2008	Barcelona	234.00	12/06/2008	13/06/2008	Expo Hotel Barcelona	110.00	10.1
MSVVEIS 2008	Barcelona	234.00	12/06/2008	13/06/2008	Olivia Plaza Hotel	125.00	10.1
LID 2008	Rome	275.00	15/05/2008	16/05/2008	Welcome Residences	140.00	06.2
LID 2008	Rome	275.00	15/05/2008	16/05/2008	Ariston	149.00	06.2
LID 2008	Rome	275.00	15/05/2008	16/05/2008	Prime Hotel Principe Torl	170.00	06.2
LID 2008	Rome	275.00	15/05/2008	16/05/2008	Giulio Cesare	185.00	06.2
LID 2008	Rome	275.00	15/05/2008	16/05/2008	Starhotels Metropole	230.00	06.2
RCIS'08	Marrakech	467.00	03/06/2008	06/06/2008	Le Meridien N'fis	132.00	11.4
RCIS'08	Marrakech	467.00	03/06/2008	06/06/2008	Sofitel Marrakech	135.00	11.4
RCIS'08	Marrakech	467.00	03/06/2008	06/06/2008	Palmeraie Golf Palace	210.00	11.4
RCIS'08	Marrakech	467.00	03/06/2008	06/06/2008	Palmeraie Village	252.00	11.4
RCIS'08	Marrakech	467.00	03/06/2008	06/06/2008	Coralia Club Palmariva M	267.00	11.4

THANK YOU!

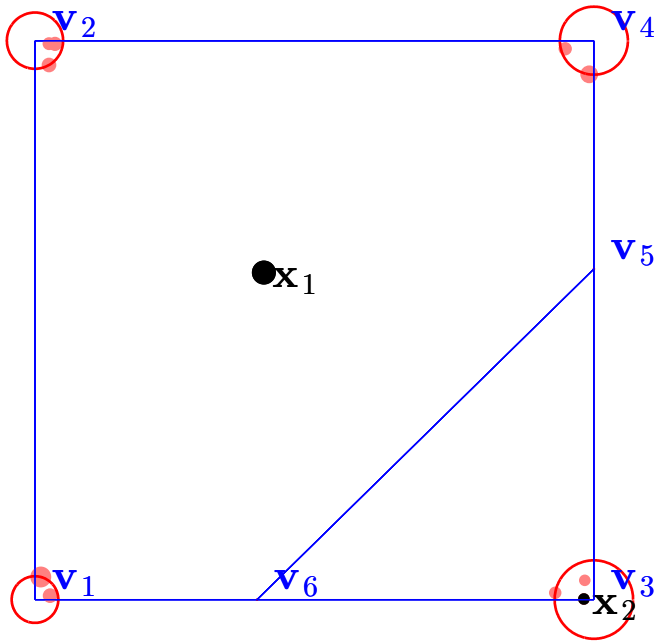
Example of diversification

- Inside red circumferences: explored region
- Pink discs: objects retrieved by distance-based access



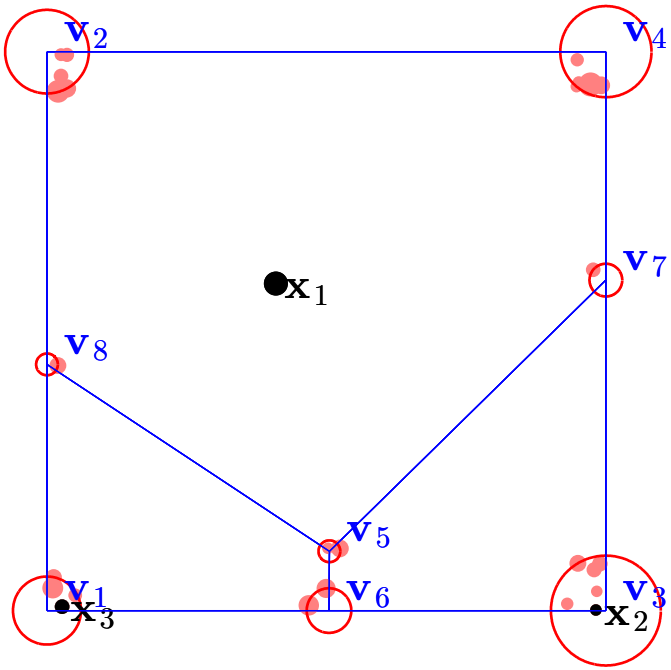
Example of diversification

- Inside red circumferences: explored region
- Pink discs: objects retrieved by distance-based access



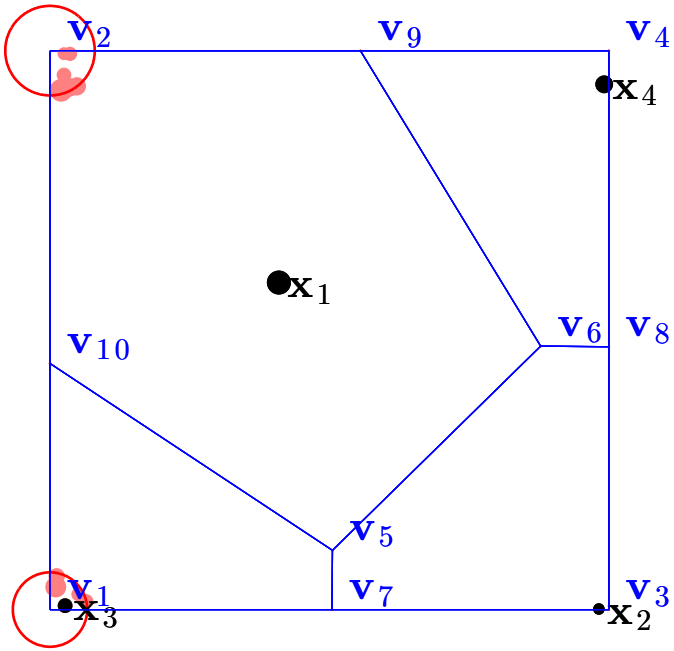
Example of diversification

- Inside red circumferences: explored region
- Pink discs: objects retrieved by distance-based access



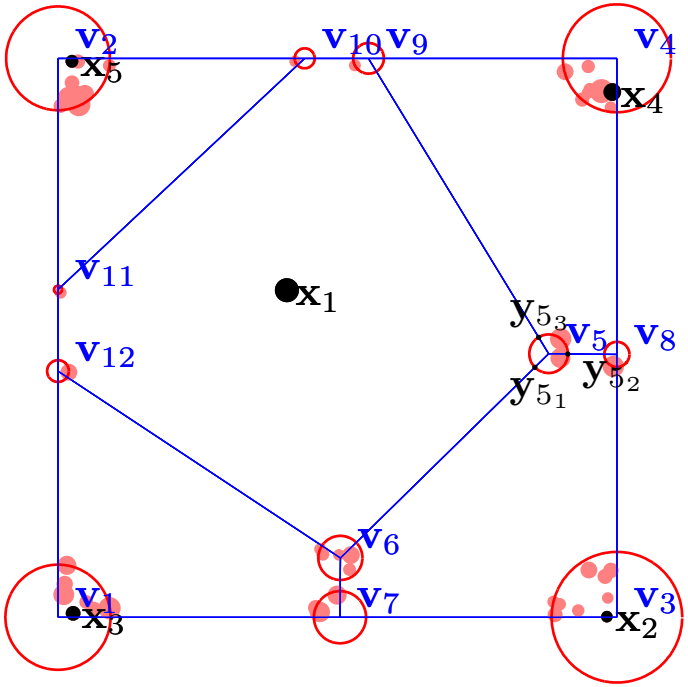
Example of diversification

- Inside red circumferences: explored region
- Pink discs: objects retrieved by distance-based access



Example of diversification

- Inside red circumferences: explored region
- Pink discs: objects retrieved by distance-based access



Main References

Historical papers

- Jean-Charles de Borda
  Mémoire sur les élections au scrutin. Histoire de l'Académie Royale des Sciences, Paris 1781
- Nicolas de Condorcet
  Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix, 1785
- Kenneth J. Arrow
  A Difficulty in the Concept of Social Welfare. Journal of Political Economy. 58 (4): 328–346, 1950

Rank aggregation and ranking queries

- Ronald Fagin, Ravi Kumar, D. Sivakumar
  Efficient similarity search and classification via rank aggregation. SIGMOD Conference 2003: 301-312
- Ronald Fagin
  Combining Fuzzy Information from Multiple Systems. PODS 1996: 216-226
- Ronald Fagin
  Fuzzy Queries in Multimedia Database Systems. PODS 1998: 1-10
- Ronald Fagin, Amnon Lotem, Moni Naor
  Optimal Aggregation Algorithms for Middleware. PODS 2001

Skylines and k-Skybands

- Stephan Börzsönyi, Donald Kossmann, Konrad Stocker
  The Skyline Operator. ICDE 2001: 421-430
- Jan Chomicki, Parke Godfrey, Jarek Gryz, Dongming Liang
  Skyline with Presorting. ICDE 2003: 717-719
- Dimitris Papadias, Yufei Tao, Greg Fu, Bernhard Seeger
  Progressive skyline computation in database systems. ACM Trans. Database Syst. 30(1): 41-82 (2005)

Main References

Extensions of skylines: flexible skylines, ORD/ORU

- Paolo Ciaccia, Davide Martinenghi
  Reconciling Skyline and Ranking Queries. PVLDB 10(11): 1454-1465 (2017)
- Paolo Ciaccia, Davide Martinenghi
  FA + TA < FSA: Flexible Score Aggregation. CIKM 2018: 57-66

Extensions of ranking queries: uncertainty, proximity, diversity

- Mohamed A. Soliman, Ihab F. Ilyas, Davide Martinenghi, Marco Tagliasacchi
  Ranking with uncertain scoring functions: semantics and sensitivity measures. SIGMOD Conference 2011: 805-816
- Davide Martinenghi, Marco Tagliasacchi
  Proximity Rank Join. PVLDB 3(1): 352-363 (2010)
- Piero Fraternali, Davide Martinenghi, Marco Tagliasacchi
  Top-k bounded diversification. SIGMOD Conference 2012: 421-432
- Akrivi Vlachou, Christos Doulkeridis, Yannis Kotidis, Kjetil Nørvåg
  Reverse top-k queries. ICDE 2010: 365-376
- Davide Martinenghi, Marco Tagliasacchi:
  Cost-Aware Rank Join with Random and Sorted Access. IEEE Trans. Knowl. Data Eng. 24(12): 2143-2155 (2012)
- Davide Martinenghi, Marco Tagliasacchi:
  Proximity measures for rank join. ACM Trans. Database Syst. 37(1): 2:1-2:46 (2012)
- Piero Fraternali, Davide Martinenghi, Marco Tagliasacchi:
  Efficient Diversification of Top-k Queries over Bounded Regions. SEBD 2012: 139-146
- Ilio Catallo, Eleonora Ciceri, Piero Fraternali, Davide Martinenghi, Marco Tagliasacchi:
  Top-k diversity queries over bounded regions. ACM Trans. Database Syst. 38(2): 10 (2013)

Main References

Web Access

- Daniele Braga, Stefano Ceri, Florian Daniel, Davide Martinenghi:
  Optimization of multi-domain queries on the web. Proc. VLDB Endow. 1(1): 562-573 (2008)
- Andrea Calì, Davide Martinenghi:
  Conjunctive Query Containment under Access Limitations. ER 2008: 326-340
- Andrea Calì, Davide Martinenghi:
  Querying Data under Access Limitations. ICDE 2008: 50-59
- Andrea Calì, Diego Calvanese, Davide Martinenghi:
  Dynamic Query Optimization under Access Limitations and Dependencies. J. Univers. Comput. Sci. 15(1): 33-62 (2009)
- Andrea Calì, Davide Martinenghi:
  Optimizing Query Processing for the Hidden Web. APWeb 2010: 397
- Andrea Calì, Davide Martinenghi:
  Querying the deep web. EDBT 2010: 724-727