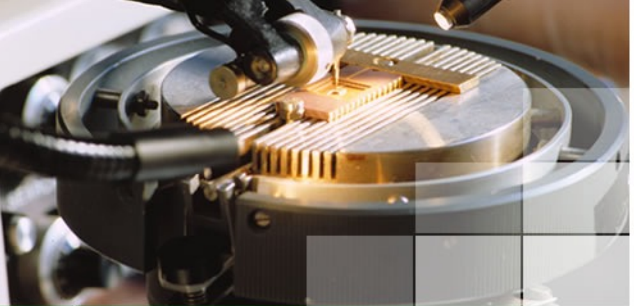


POLITECNICO DI MILANO

Dipartimento di
Elettronica e Informazione

Reducing Uncertainty in Top-K Queries

Davide Martinenghi

Joint work with I. Catallo, E. Ciceri, P. Fraternali, and M. Tagliasacchi

Trento, November 21, 2013

Outline

- Rank aggregation and rank join
- Uncertain scoring
- Representative orderings
- Reducing uncertainty through human workers

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

[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 means of 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

[Vlachou et al., ICDE 2010]

- Aggregation function:

$$\text{Score(cand)} = w_1\ s_1 + w_2\ s_2 + w_3\ s_3 + w_4\ s_4 + w_5\ s_5$$

Cand	s_1	Cand	s_2	Cand	s_3	Cand	s_4	Cand	s_5
1	.9	2	.65	4	.99	5	.6	3	.8
2	.7	1	.6	2	.97	1	.5	1	.7
3	.5	5	.55	5	.95	3	.4	5	.65
4	.3	4	.5	3	.93	4	.3	2	.63
5	.1	3	.45	1	.91	2	.2	4	.62

Judge 1 **Judge 2** **Judge 3** **Judge 4** **Judge 5**

- What weights should I convince you to use so that my preferred candidate becomes the best?
  - (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

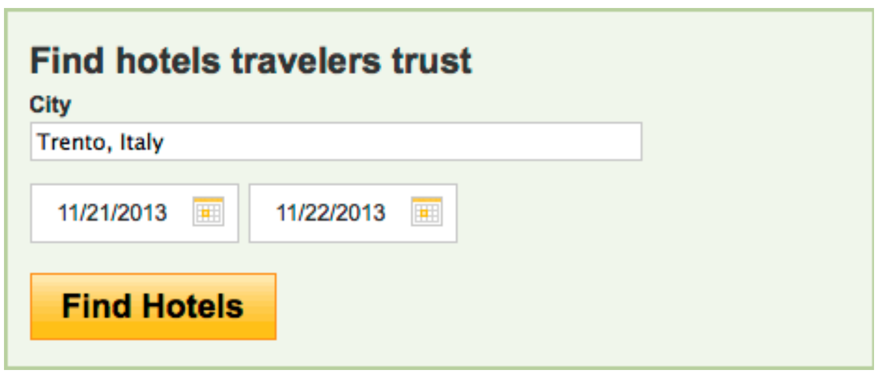
Ranking in the real world

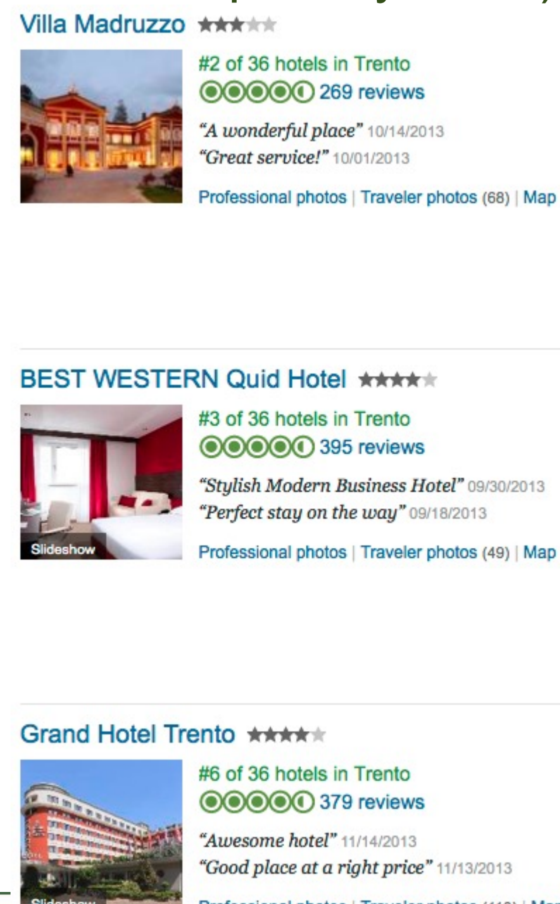
[Calì & Martinenghi, ICDE 2008] [Martinenghi & Tagliasacchi, TKDE 2012]

- 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$^{o,\text{ranked}}$)

  - Other needs: joins (rank join)
  - But also: dirty data, deduplication, diversification, uncertainty, incompleteness, recency, paging, access costs…



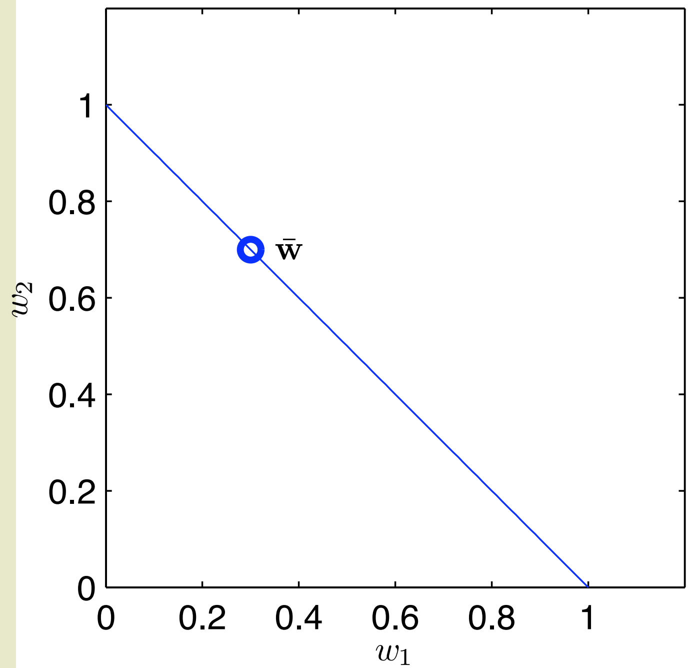


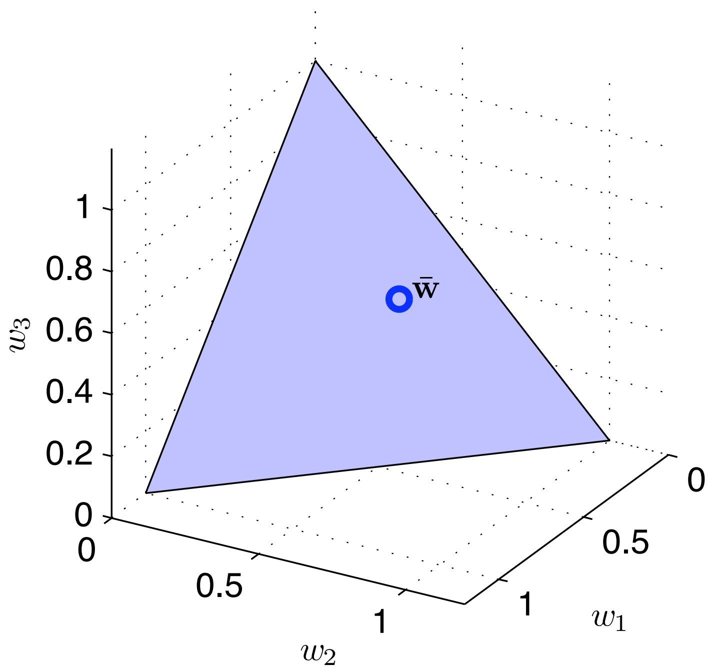
Uncertain scoring

[Soliman & Ilyas, ICDE 2009], [Soliman et al., SIGMOD 2011]

- Users are often unable to precisely specify the scoring function
- Objects may have imprecise scores, e.g., defined over intervals
  - E.g., apartment rent [$200-$250]
- Using trial-and-error or machine learning may be tedious and time consuming
- Even when the function is known, it is crucial to analyze the sensitivity of the computed ordering wrt. changes in the function

- Assumptions:
  - Linear scoring function: $S = w_1s_1 + \ldots + w_ns_n$
  - User-defined weights $w_1,\ldots,w_n$ are uncertain, and, w.l.o.g., normalized to sum up to 1
- Each point on the simplex represents a possible scoring function





Example

- Top-k query:

SELECT R.RestName, R.Street, H.HotelName
FROM RestaurantsInParis R, HotelsInParis H
WHERE distance(R.coordinates, H.coordinates) $\leq 500m$
RANK BY $w_R \cdot$ R.Rating $+ \; w_H \cdot$ H.Stars
LIMIT 5

- Results and possible orderings:

ID	rating	stars
τ_1	2	6
τ_2	7	5
τ_3	4	7
τ_4	5	2

Join Results

Rank By $w_R.rating+w_H.stars$

$w_R+w_H=1$

λ^1	λ^2	λ^3	λ^4	λ^5
τ_3	τ_3	τ_2	τ_2	τ_2
τ_1	τ_2	τ_3	τ_3	τ_4
τ_2	τ_1	τ_1	τ_4	τ_3
τ_4	τ_4	τ_4	τ_1	τ_1

w_R: 0 0.167 0.4 0.571 0.833 1.0

Representative ordering

- Both value uncertainty and weight uncertainty determine score uncertainty
  - This induces a partial order over objects
  - we have a space of possible orderings
- We focus on a representative of the space
- An example is the Most Probable Ordering

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

- Other definitions of representative ordering exist, e.g., the Optimal Rank Aggregation

Example of MPO

- For K=2, the MPO is <τ_2, τ_3>
  - under the assumption of uniform probability distribution

ID	rating	stars
τ_1	2	6
τ_2	7	5
τ_3	4	7
τ_4	5	2

Join Results

Rank By $w_R.rating+w_H.stars$

$w_R+w_H=1$

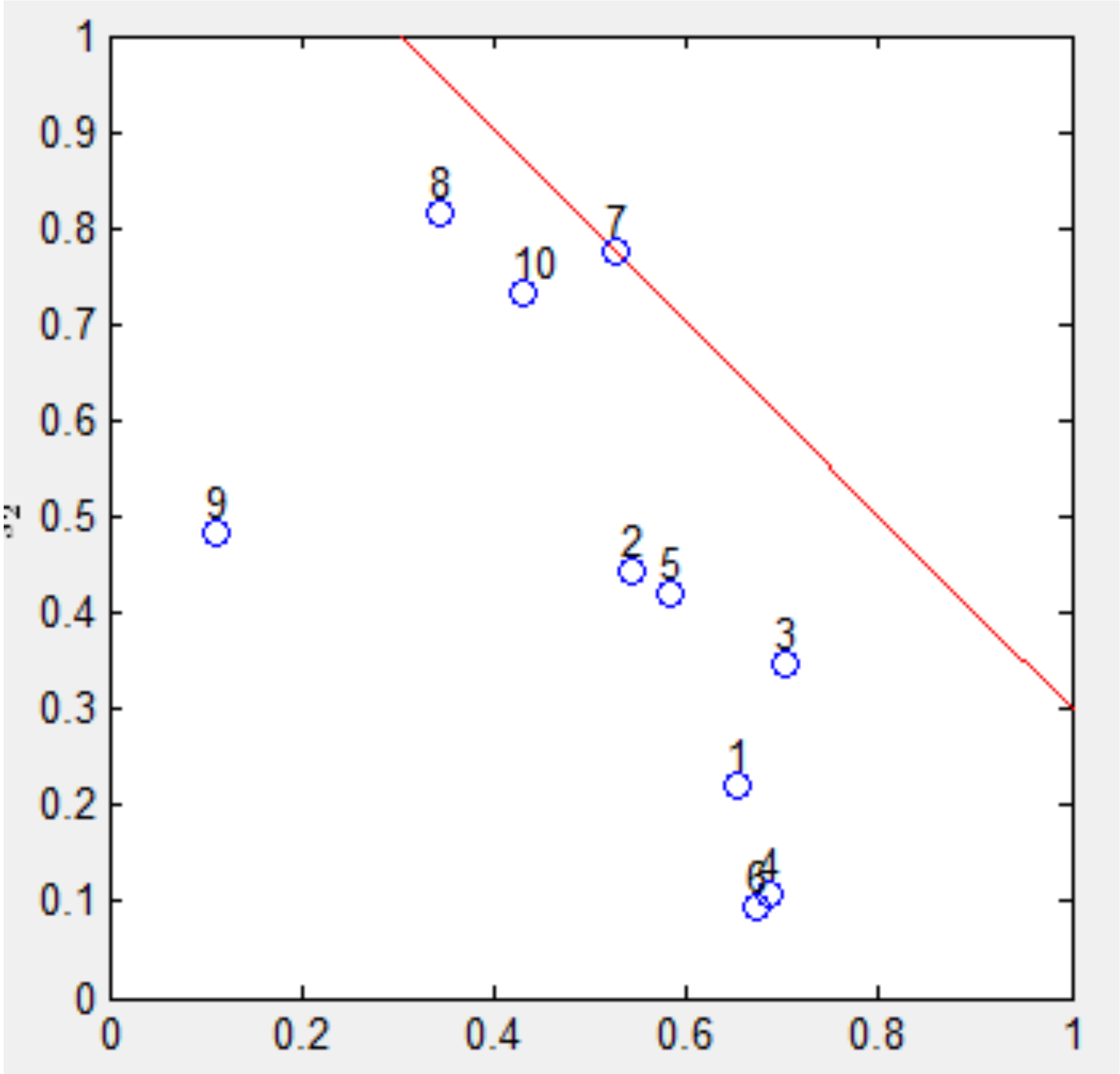
λ^1	λ^2	λ^3	λ^4	λ^5
τ_3	τ_3	τ_2	τ_2	τ_2
τ_1	τ_2	τ_3	τ_3	τ_4
τ_2	τ_1	τ_1	τ_4	τ_3
τ_4	τ_4	τ_4	τ_1	τ_1

w_R: 0 — 0.167 — 0.4 — 0.571 — 0.833 — 1.0

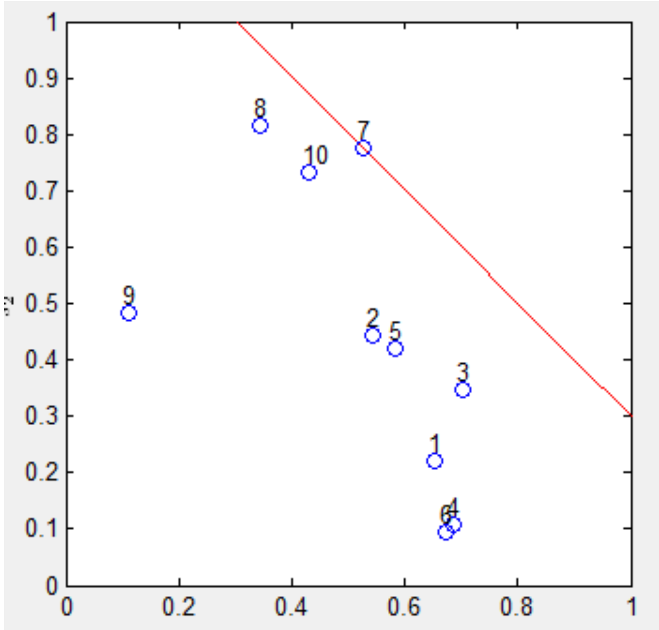
Shortcomings of representative orderings

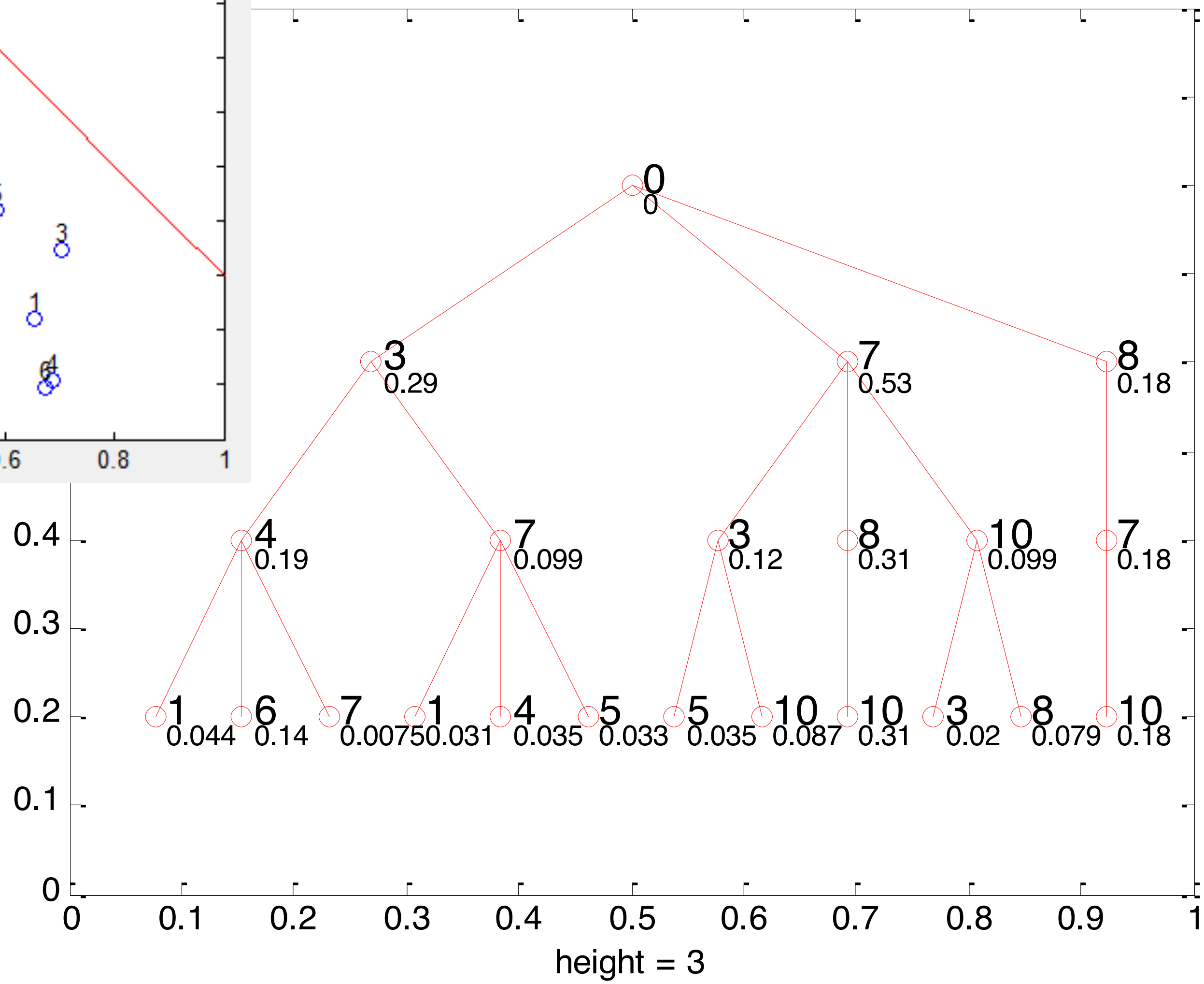
- Complex to compute:
  - exponential in the number of dimensions (weights)
  - in some cases, NP-hard already in 3D
- MPOs may fail to be truly representative:
  - often, only slightly better than the second most probable ordering
  - how stable is the ordering? would it remain the same after a slight perturbation of the weights?

Points corresponding to join results for d=2



Construction of tree of possible orderings





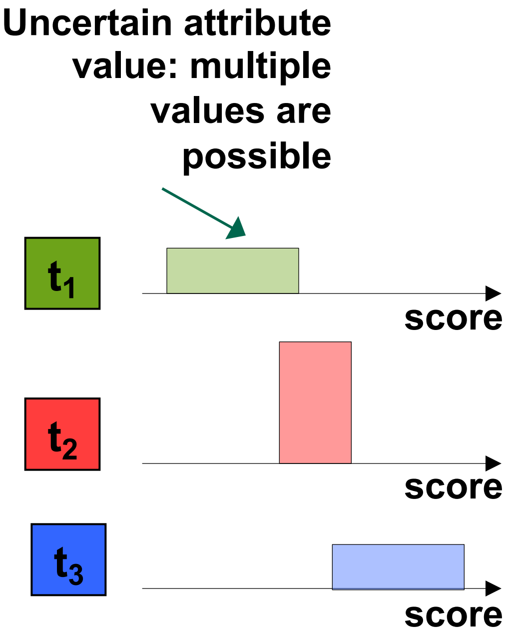
Asking humans

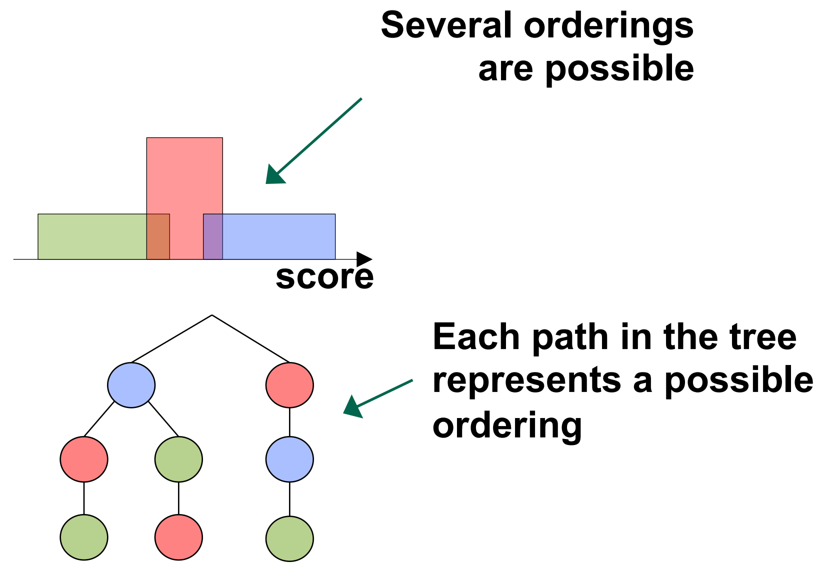
- Question answering:
  - How to use human workers to reduce the amount of uncertainty?
  - Which questions to pose?
- Task assignment:
  - Once the tasks are defined, which humans to ask?

Uncertainty reduction via question answering

[Li & Deshpande, VLDB 2010]

- When several orderings are possible, the space of possible orderings compatible with the score values can be determined and represented as a tree
- Each node is associated with a probability





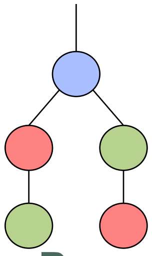
Uncertainty reduction via question answering

Determining the best ordering

REQUIRES TO

Reduce uncertainty in the space of possible orderings

=



Prune paths

Solution: crowdsourcing

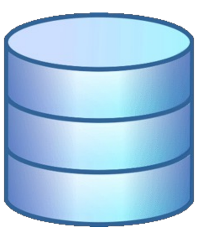
Questions

Crowd

User knowledge

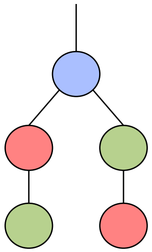
1) **Resolve conflicts** (i.e., ambiguities on the ordering of two or more objects)

2) **Refine score intervals**



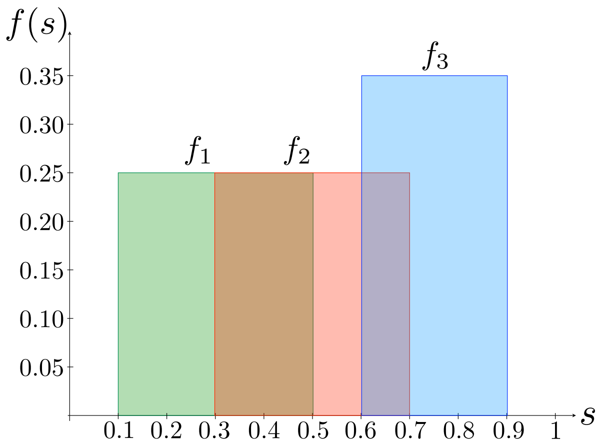
Reduce uncertainty in the space of possible orderings

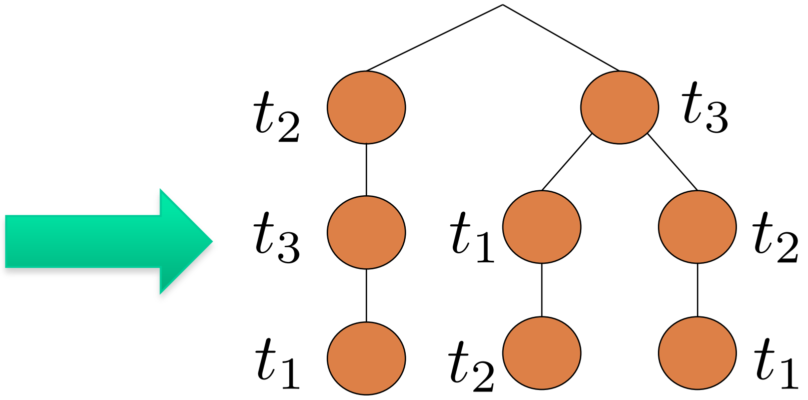
=



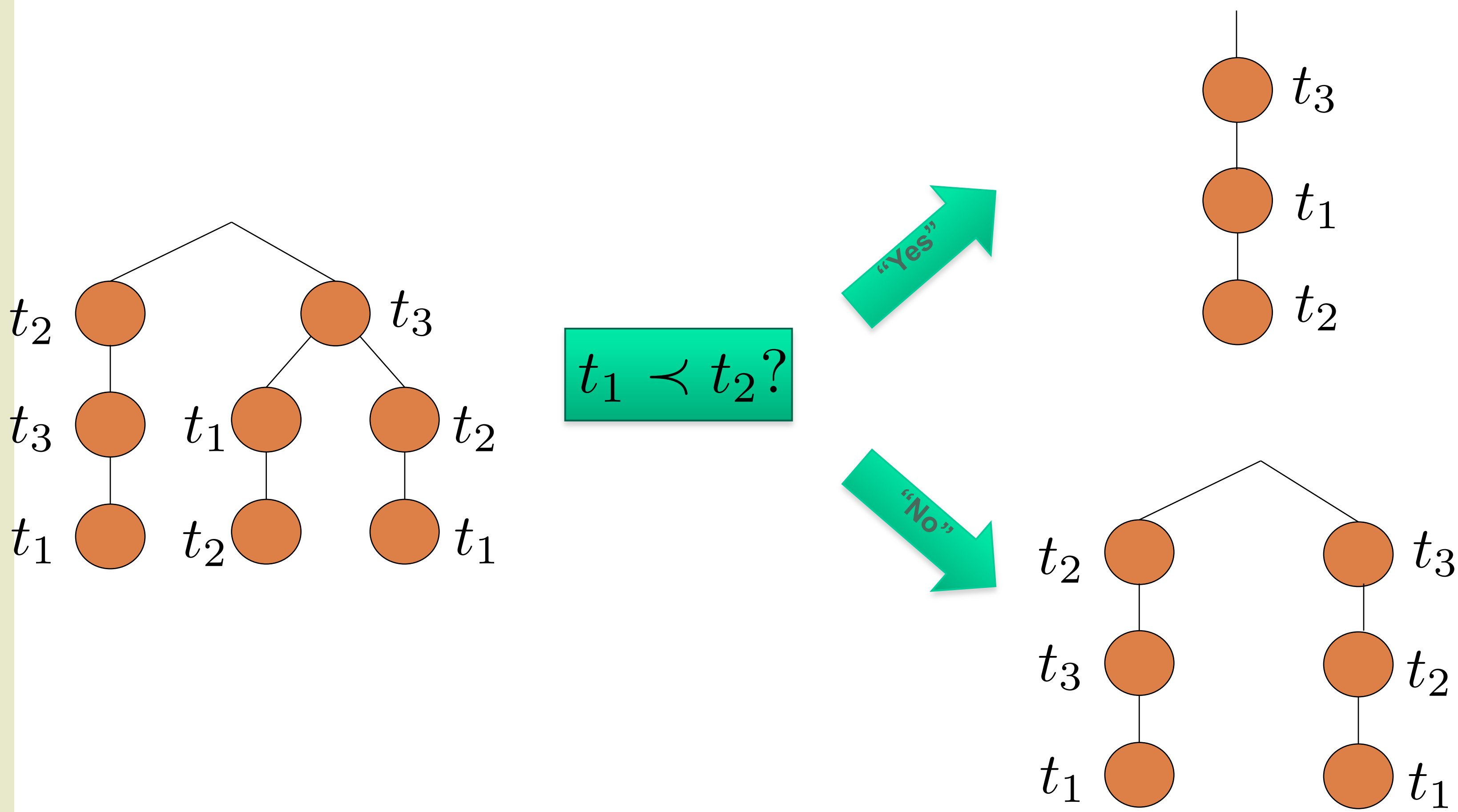
Prune paths

Showcase: tree construction





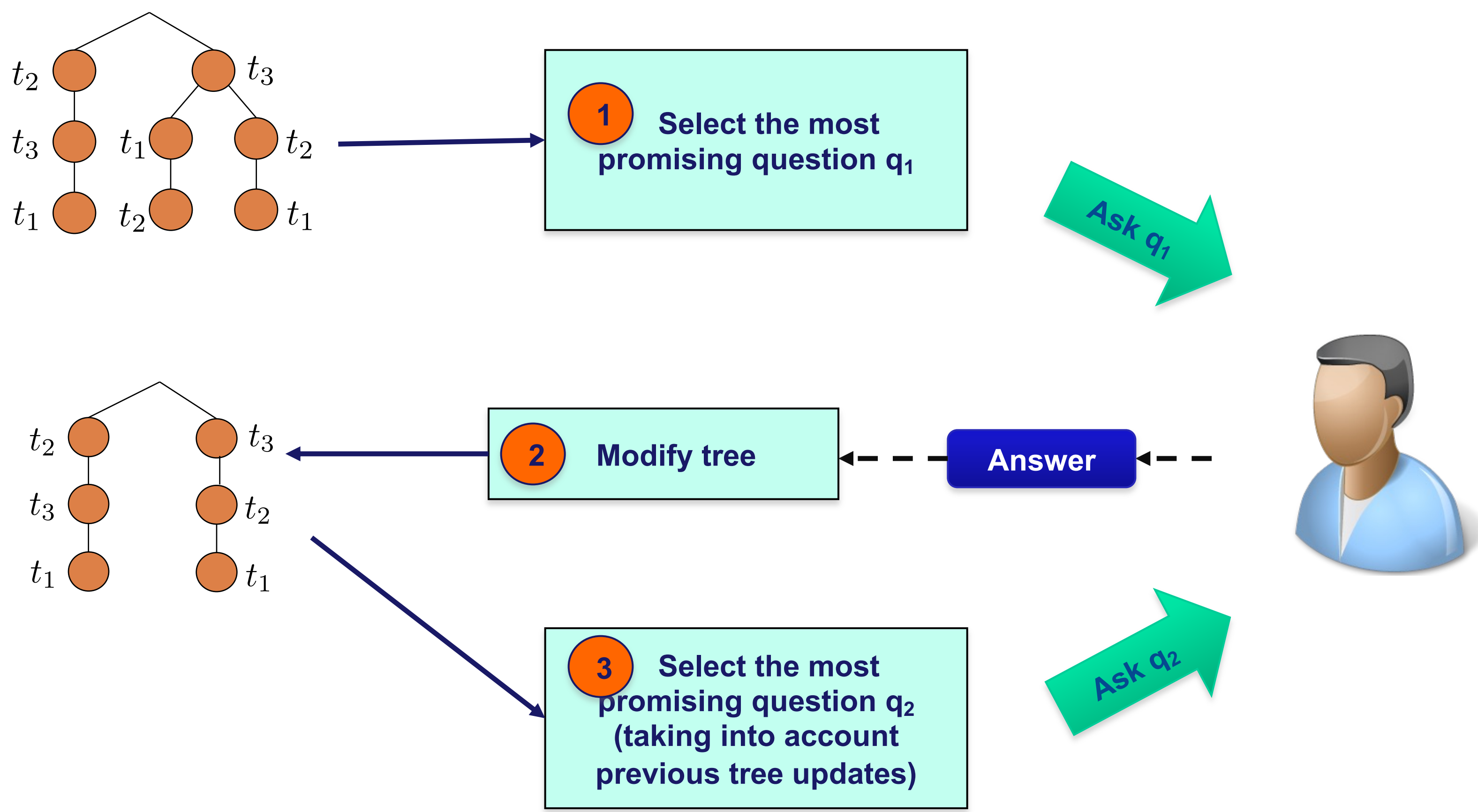
Showcase: question answering



Open issue: question types

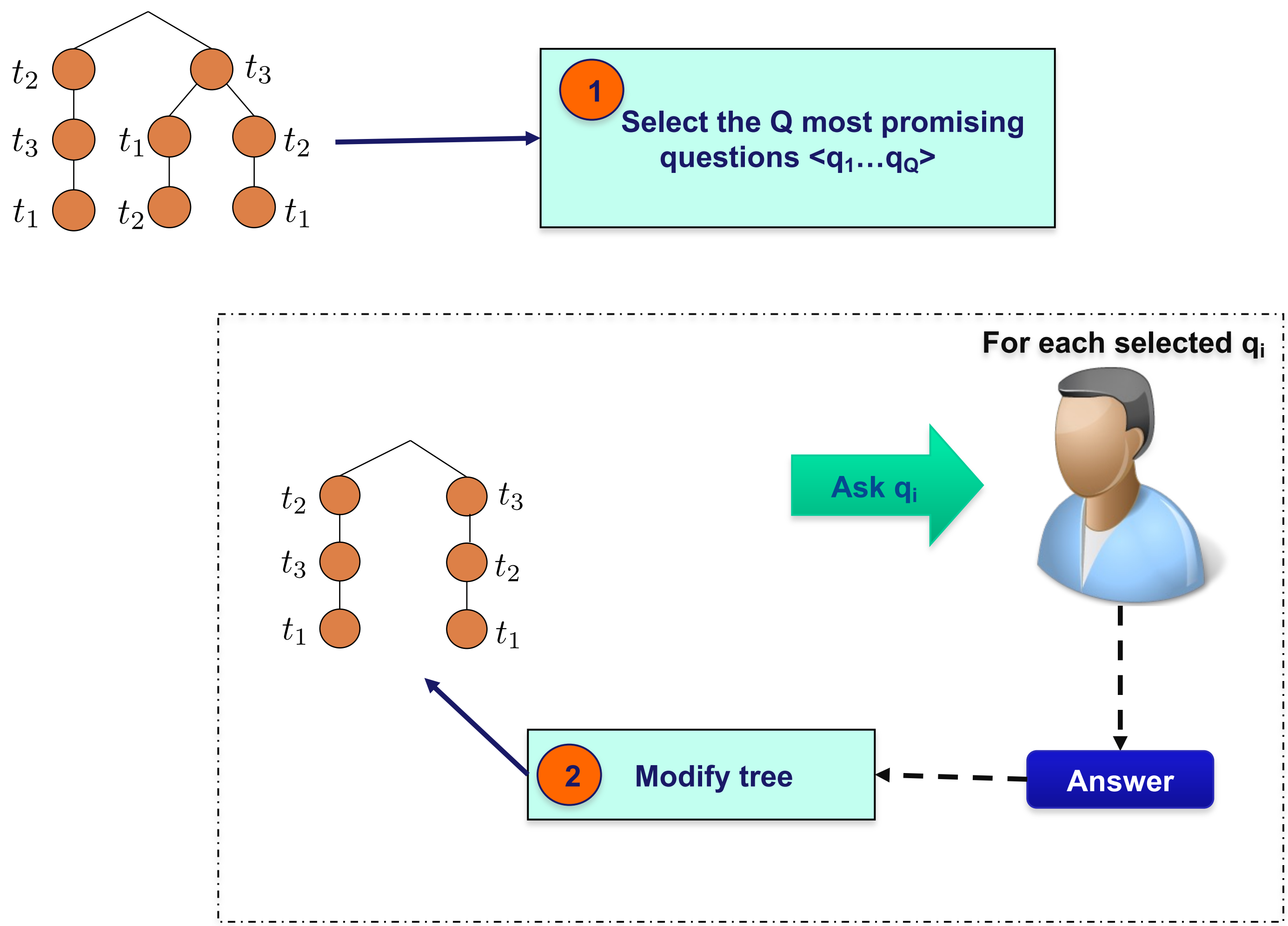
- Questions
  - Define the types of possible questions
  - Define how to measure uncertainty in the space of possible orderings, so as to check its reduction as questions are answered
- Measuring uncertainty
  - Shannon's entropy (or some discounted version thereof)
  - Distance from a representative ordering
  - …
- Uncertainty reduction
  - Devise the optimal set/sequence of Q questions that can be posed to users

First solution: Online approach



…

Second solution: Offline approach



Comparison

	Online Approach	Offline approach
PROS	Optimized with respect to the **actual system state**	**Fast** user interaction (questions are chosen before interacting with the user)
CONS	**Slow** user interaction (questions are evaluated at each step)	Questions are chosen according to the **initial system state** (+some clues about the future gains), not according to the system state at each step

Crowdsourcing marketplaces

- **Crowdsourcing marketplaces:** Internet marketplaces that enable requesters to hire crowd workers to perform tasks

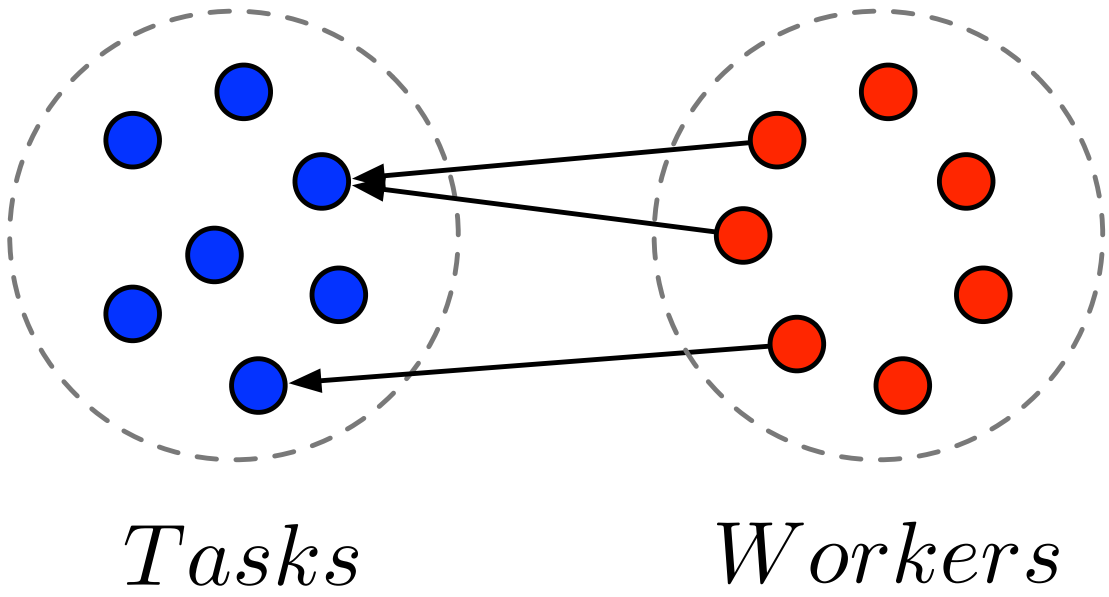
Task assignment: Motivations

[Raykar et al., J. of Machine Learning Research 2010]

- It is often the case that a worker does not have the **appropriate knowledge** for annotating all the data, even for a particular domain
- Each worker is characterized by different parameters we should take into consideration
- Examples:
  - Expertise
  - Geocultural information
  - Past work history
- **Problem:** How to associate the most suitable task with the most appropriate worker(s)?

Task assignment: Definition

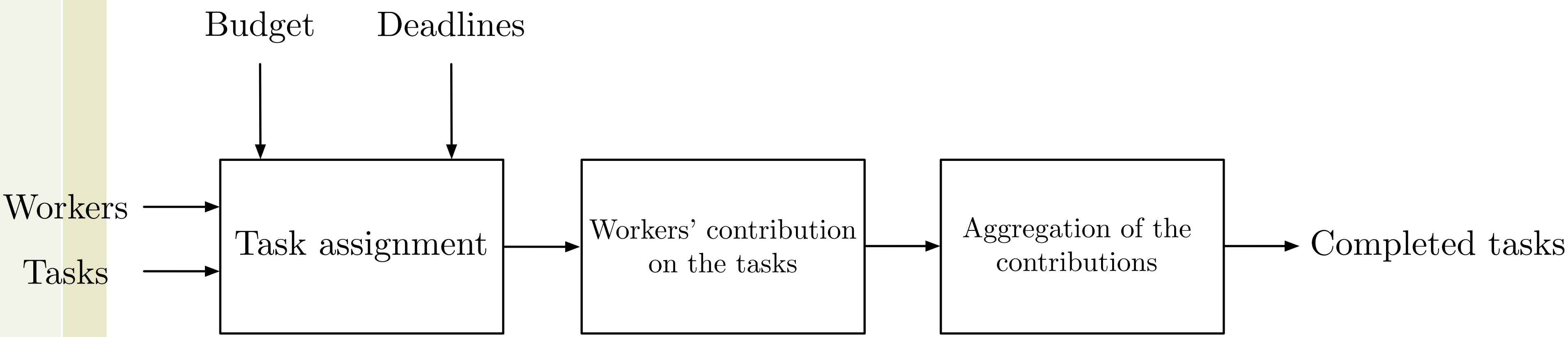
- **Task assignment:** identify the best assignment configuration between workers and tasks, given an upper bound on the *number of assignments or a delay constraint* (i.e., *who should work on what?)*
- Expressed by means of a bipartite assignment graph
- Constrained maximization problem (maximize assignment quality over all feasible task-annotator assignments)



Objectives and parameters

- Parameters of interest:
  - **Worker model:** accuracy (probability of correctly solving the task), fatigue decay, cost, correlation
  - **Task model:** uncertainty
- Optimal allocation
  - **Possible objectives:**
    - Achieving maximum quality given a target budget
    - Ensuring that tasks finish before a target deadline

Execution pipeline of a task assignment policy



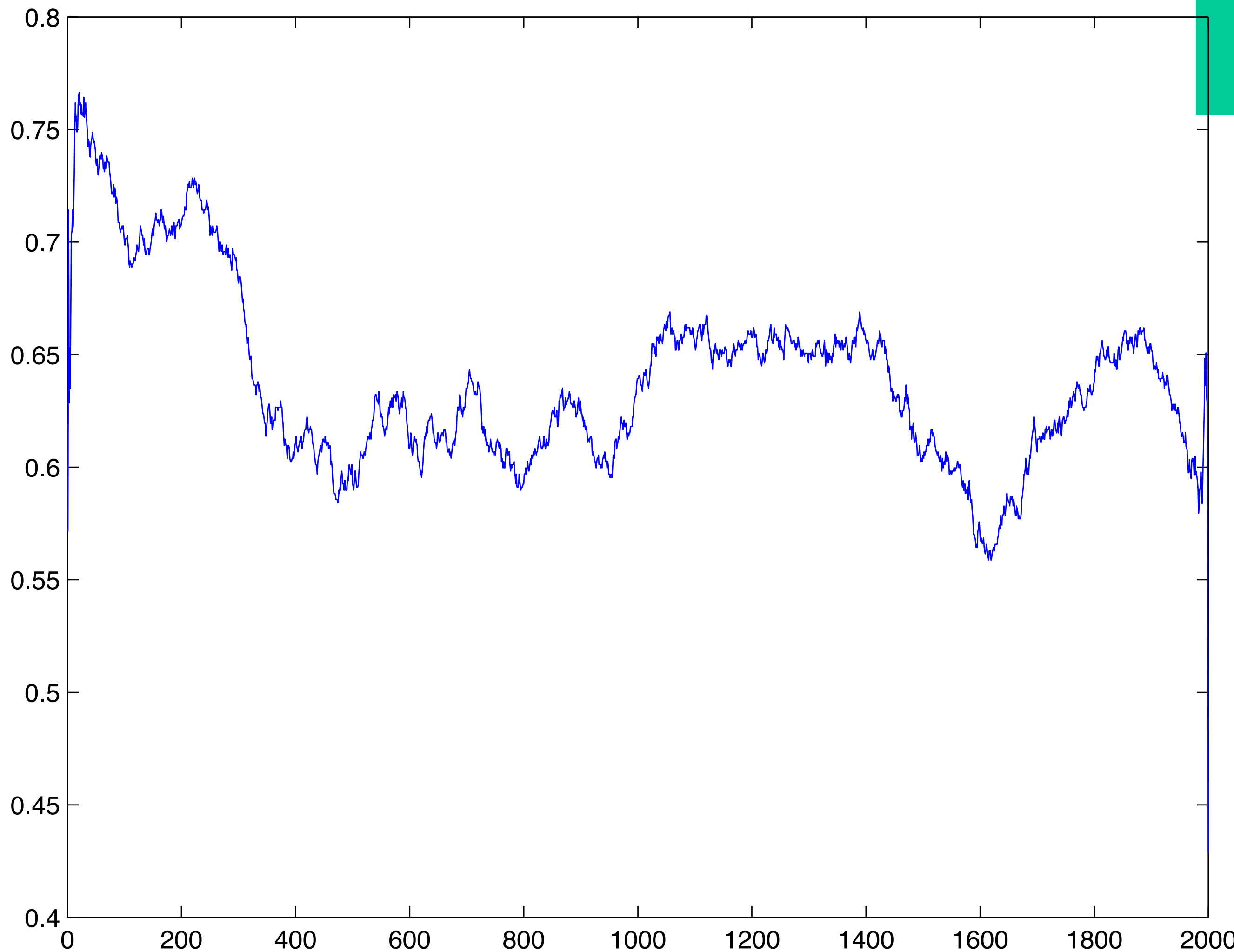
Experimental assessment

- Parameters of interest:
  - Tasks' quality and completion rate w.r.t. to workers' accuracy distributions
  - Optimal budget B^* w.r.t. expected number of workers
- Experimental assessment:
  - On publicly available data sets (e.g., UCI repository)
  - On real crowds (e.g., MicroTask)

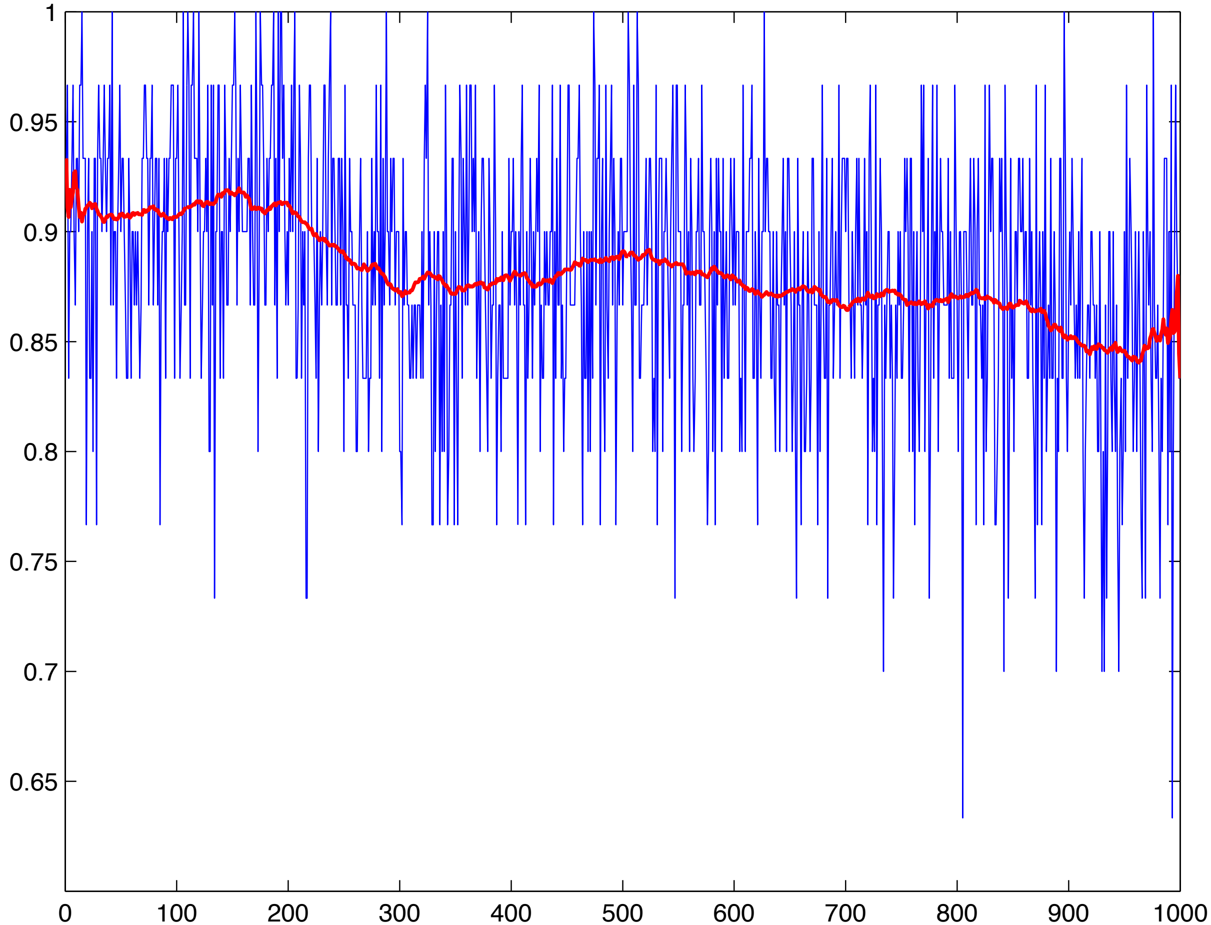
Acknowledgments: CUbRIK Project

- CUbRIK is a research project financed by the European Union
- **Goals:**
  - Advance the architecture of ***multimedia search***
  - Exploit the ***human contribution*** in multimedia search
  - Use ***open-source components*** provided by the community
  - Start up a ***search business ecosystem***
- http://www.cubrikproject.eu/

Fatigue: accuracy over time



Fatigue: accuracy with a more rewarding model



Main References

Core contributions

- Eleonora Ciceri, Piero Fraternali, Davide Martinenghi, Marco Tagliasacchi:
  Crowdsourcing for Top-K Query Processing over Uncertain Data. IEEE Trans. Knowl. Data Eng. 28(1): 41-53 (2016)
- Eleonora Ciceri, Piero Fraternali, Davide Martinenghi, Marco Tagliasacchi:
  Humans Fighting Uncertainty: Crowdsourcing for Top-K Query Processing. SEBD 2016: 78-85
- 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)
- Piero Fraternali, Davide Martinenghi, Marco Tagliasacchi:
  Top-k bounded diversification. SIGMOD Conference 2012: 421-432
- Piero Fraternali, Davide Martinenghi, Marco Tagliasacchi:
  Efficient Diversification of Top-k Queries over Bounded Regions. SEBD 2012: 139-146

Crowdsourcing applications

- Carlo Bernaschina, Ilio Catallo, Piero Fraternali, Davide Martinenghi:
  On the Role of Task Design in Crowdsourcing Campaigns. HCOMP 2015: 4-5
- Eleonora Ciceri, Ilio Catallo, Davide Martinenghi, Piero Fraternali:
  When Food Matters: Identifying Food-related Events on Twitter. KDWeb 2015: 65-76
- Carlo Bernaschina, Ilio Catallo, Piero Fraternali, Davide Martinenghi, Marco Tagliasacchi:
  Champagne: A Web Tool for the Execution of Crowdsourcing Campaigns. WWW (Companion Volume) 2015: 171-174
- Carlo Bernaschina, Piero Fraternali, Luca Galli, Davide Martinenghi, Marco Tagliasacchi:
  Robust aggregation of GWAP tracks for local image annotation. ICMR 2014: 403
- Babak Loni et al.:
  Fashion-focused creative commons social dataset. MMSys 2013: 72-77

Main References

More crowdsourcing applications

- Luca Galli, Piero Fraternali, Davide Martinenghi, Marco Tagliasacchi, Jasminko Novak:
  A Draw-and-Guess Game to Segment Images. SocialCom/PASSAT 2012: 914-917
- Alessandro Bozzon et al.:
  A Framework for Crowdsourced Multimedia Processing and Querying. CrowdSearch 2012: 42-47
- Piero Fraternali et al:
  The CUBRIK project: human-enhanced time-aware multimedia search. WWW (Companion Volume) 2012: 259-262