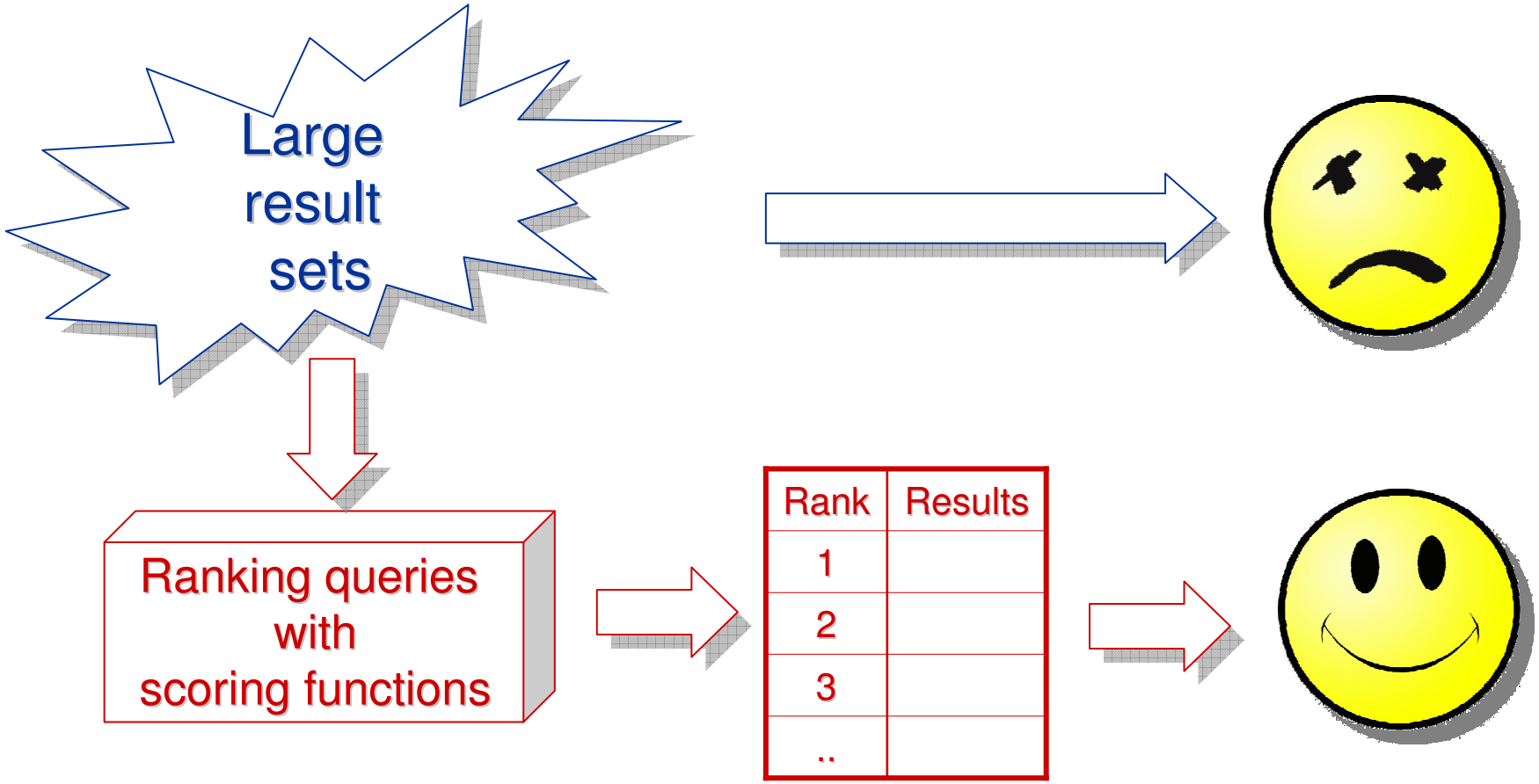
Efficiently Answering Top-k Typicality Queries on Large Databases

Ming Hua	Simon Fraser University, Canada
Jian Pei	Simon Fraser University, Canada
Ada W.C. Fu	The Chinese University of Hong Kong, China
Xuemin Lin	The University of New South Wales & NICTA, Australia
Ho-Fung Leung	The Chinese University of Hong Kong, China

Motivation



Can we rank the data according to their typicality?

Motivation Example

- *Mammals*: 5400 species, 1200 genera, 153 families and 29 order
- Which one is more typical, lion or platypus?

lion

Giving birth to live young, like most other mammals.

platypus

Laying eggs.

Challenges & Contributions

- How to define typicality in database query answering?
  - Borrow the concept from Psychology and Cognitive Science.
- How to evaluate typicality ranking queries efficiently?
  - Propose three efficient evaluation algorithms.
- How are the results on real data sets?
  - Apply typicality analysis on NBA data set and Zoo data set.

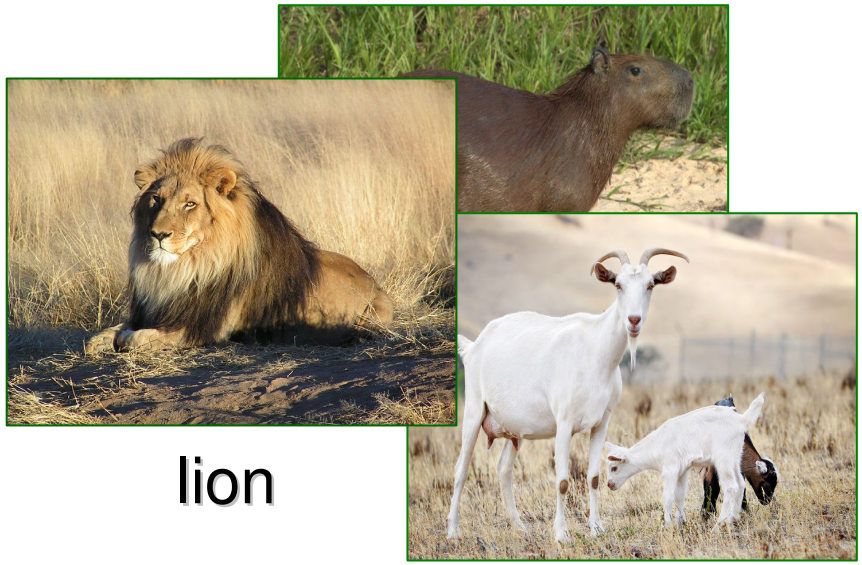
Outline

- Problem definition
  - Two types of typicality measure
- Query evaluation
  - Exact algorithm
  - Approximation algorithms
- Experimental results
  - Zoo database and NBA statistics

Psychology Point of View

"An object is more typical, if it is more similar to other objects in the same category"

"An object is more typical, if it shares more features with other objects in the same category"



lion

platypus

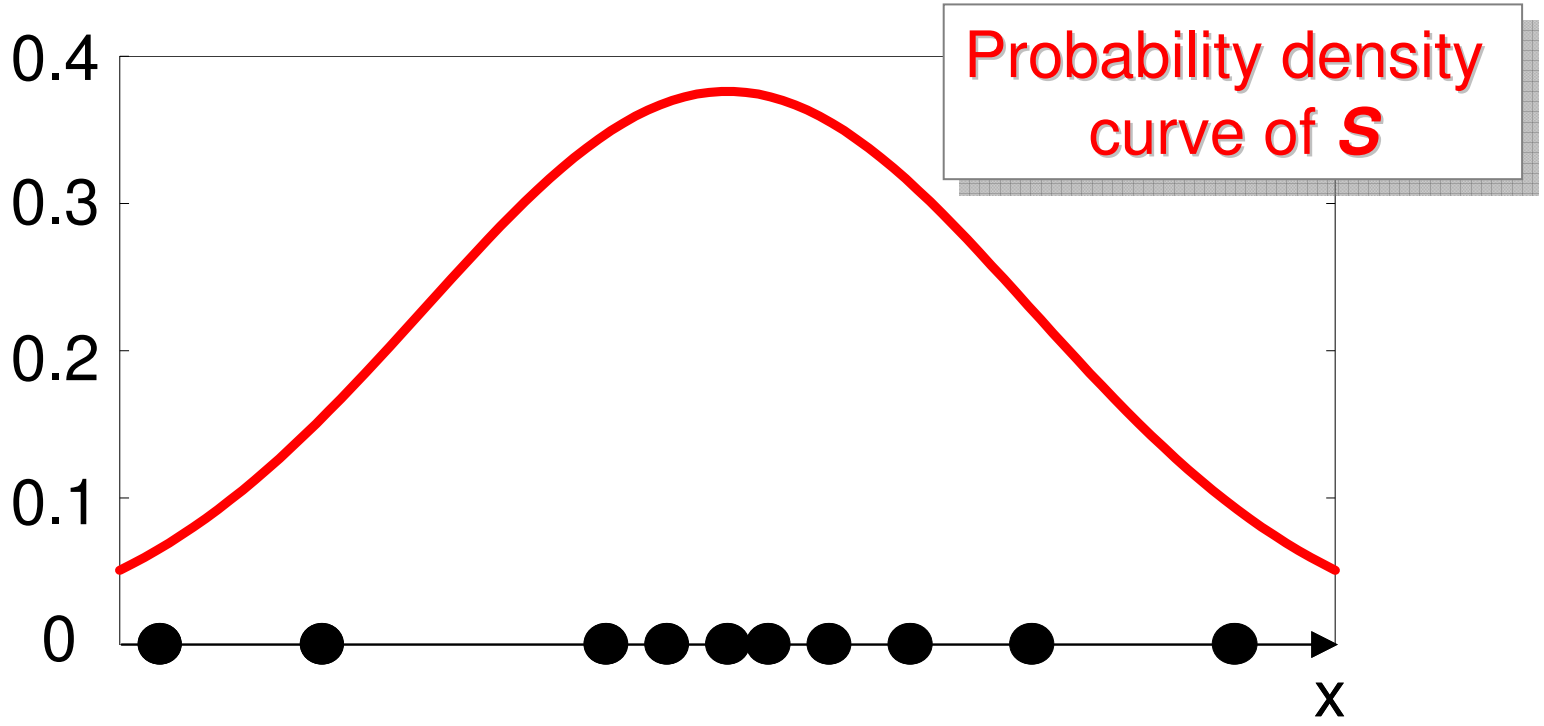
From Psychology to Database

- Lion is more typical than platypus as a mammal
- If lion and platypus are both unknown animals, lion is more likely to be labeled as a mammal

Given a set of objects S and an object o,
o is *more typical* than other objects in S,
if it is *more likely to appear* in S than others.

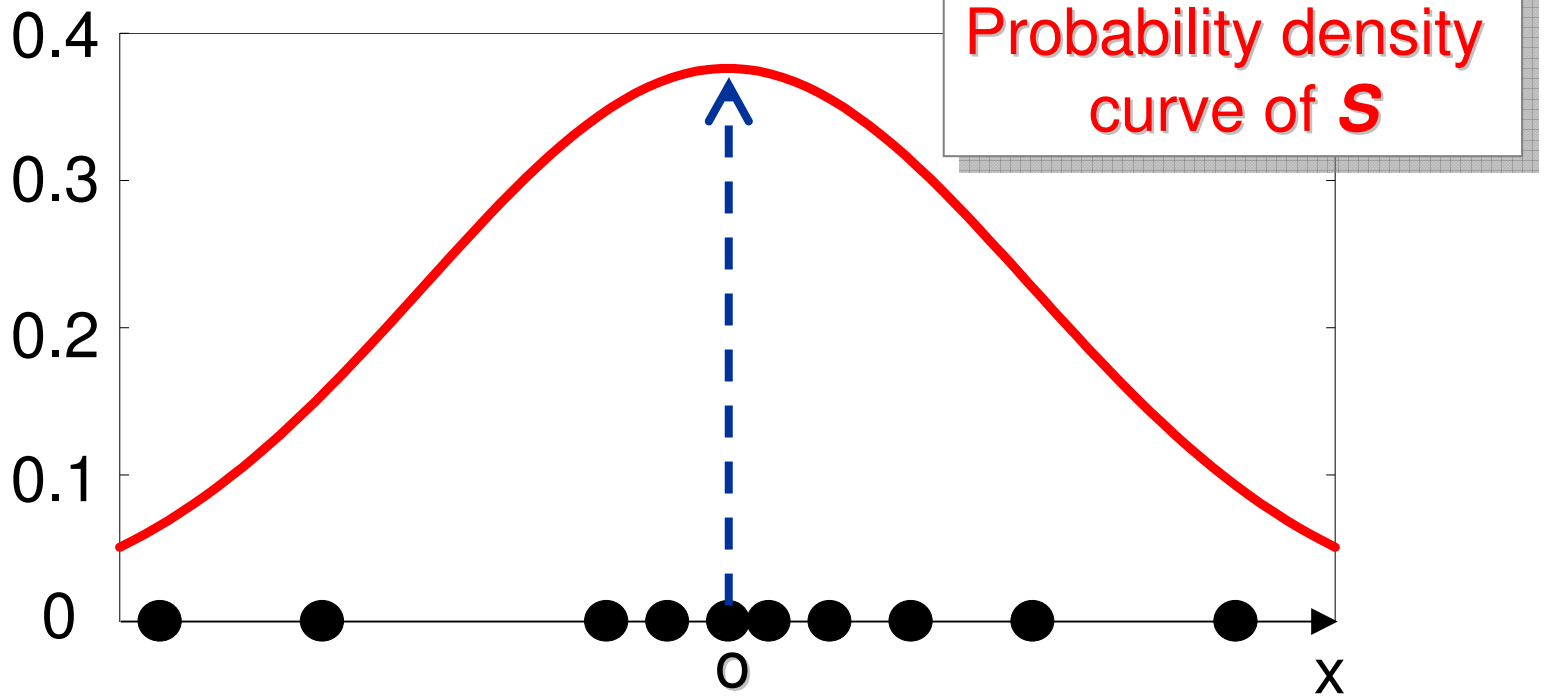
Simple Typicality

- Given a set of objects S, we treat S as an independently and identically distributed sample of a random variable ***S***.



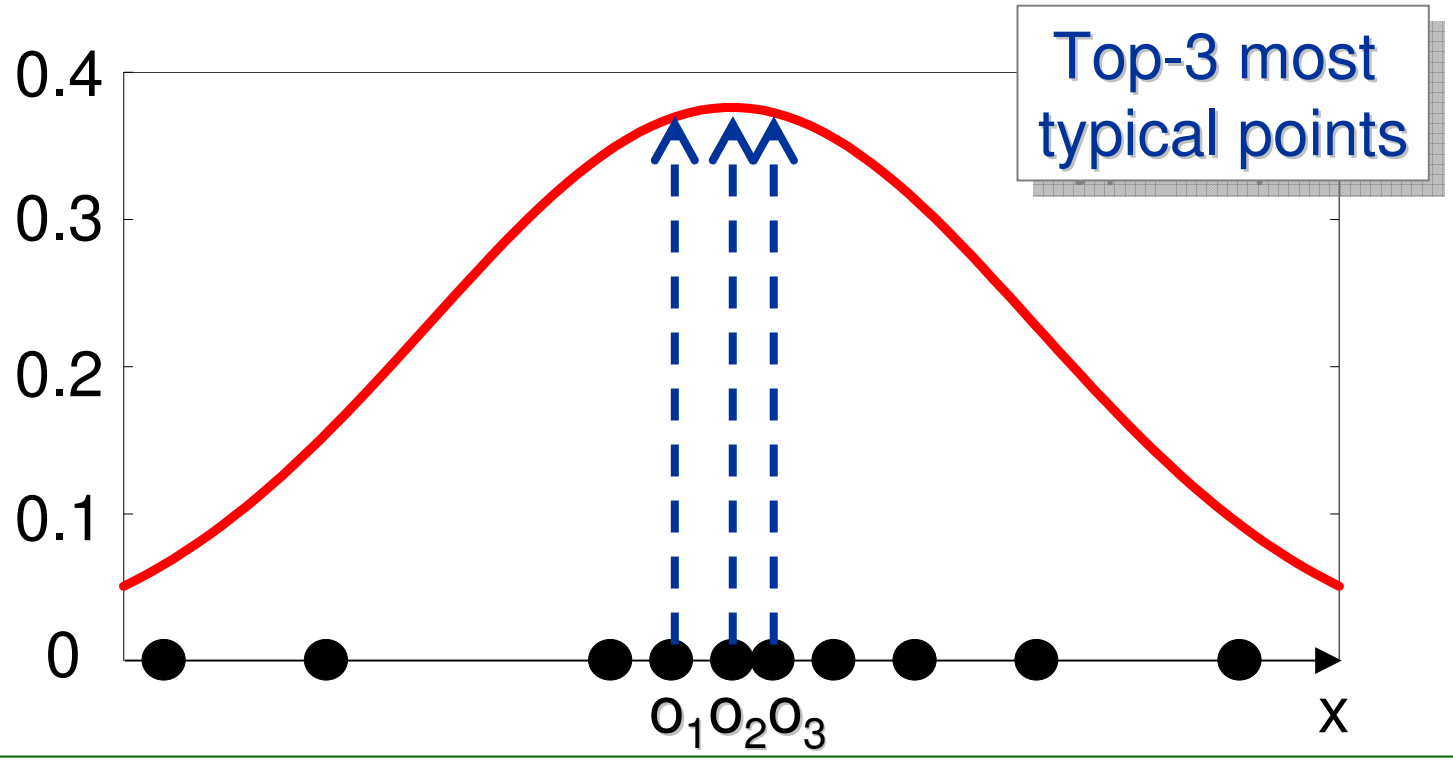
Simple Typicality

- The *simple typicality* of an object o∈S is defined as $T(o,S) = f(o)$
  - where f is the probability density function of ***S***.



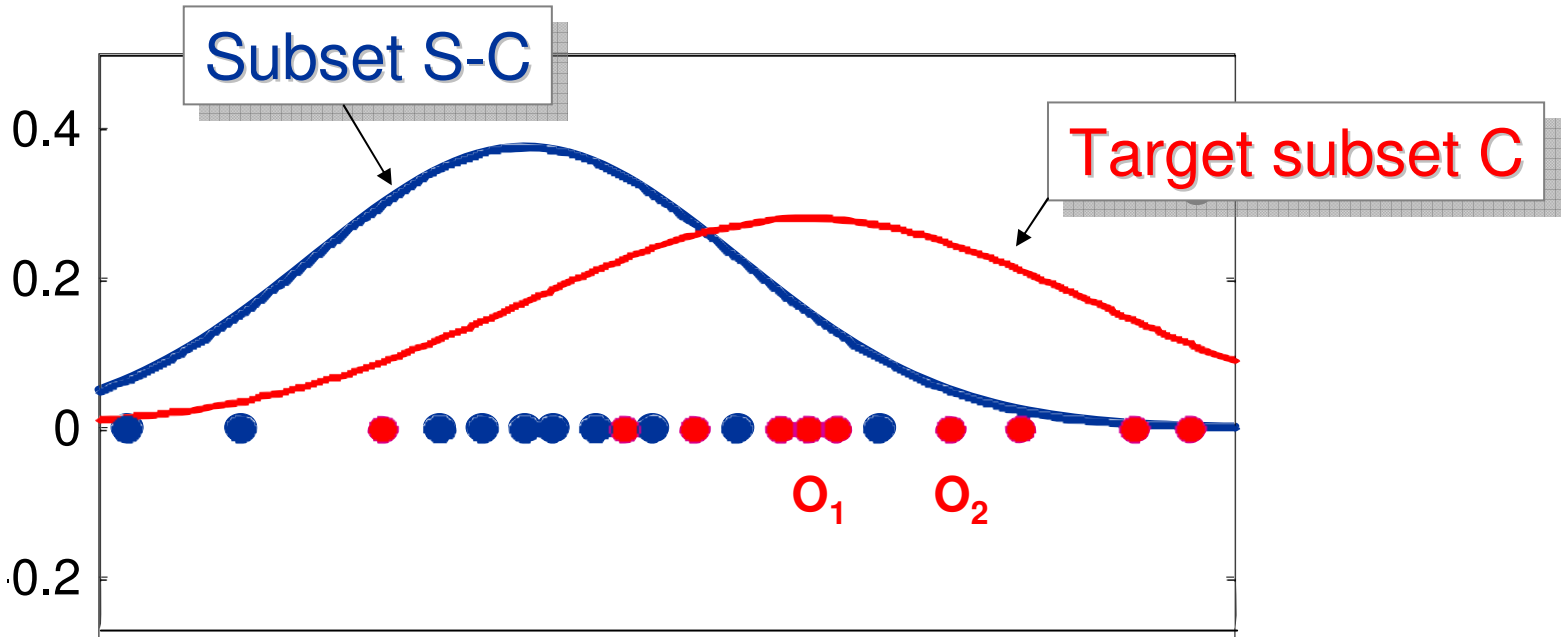
Top-k Simple Typicality Query

- A top-k simple typicality query finds the k objects with the largest simple typicality values



Discriminative Typicality

- Given a set of objects S, and a target subset C, we treat C and S-C as independently and identically distributed samples of random variable **C** and ***S-C***, respectively.



Discriminative Typicality

- The discriminative typicality of an object should capture two factors
  - How typical it is
  - How discriminative it is
- The *discriminative typicality* of an object $o \in C$ is defined as

$$DT(o, C, S) = f(o) \times [f(C \mid o) - f(S - C \mid o)]$$

Probability density of o in S.
The larger, the more typical.

Given o, the probability difference for the appearance of C and S-C.
The larger, the more discriminative.

Bayesian Explanation

- Discriminative typicality

$$DT(o,C,S) = f(o) \times [f(C \mid o) - f(S - C \mid o)]$$

Bayesian Theorem

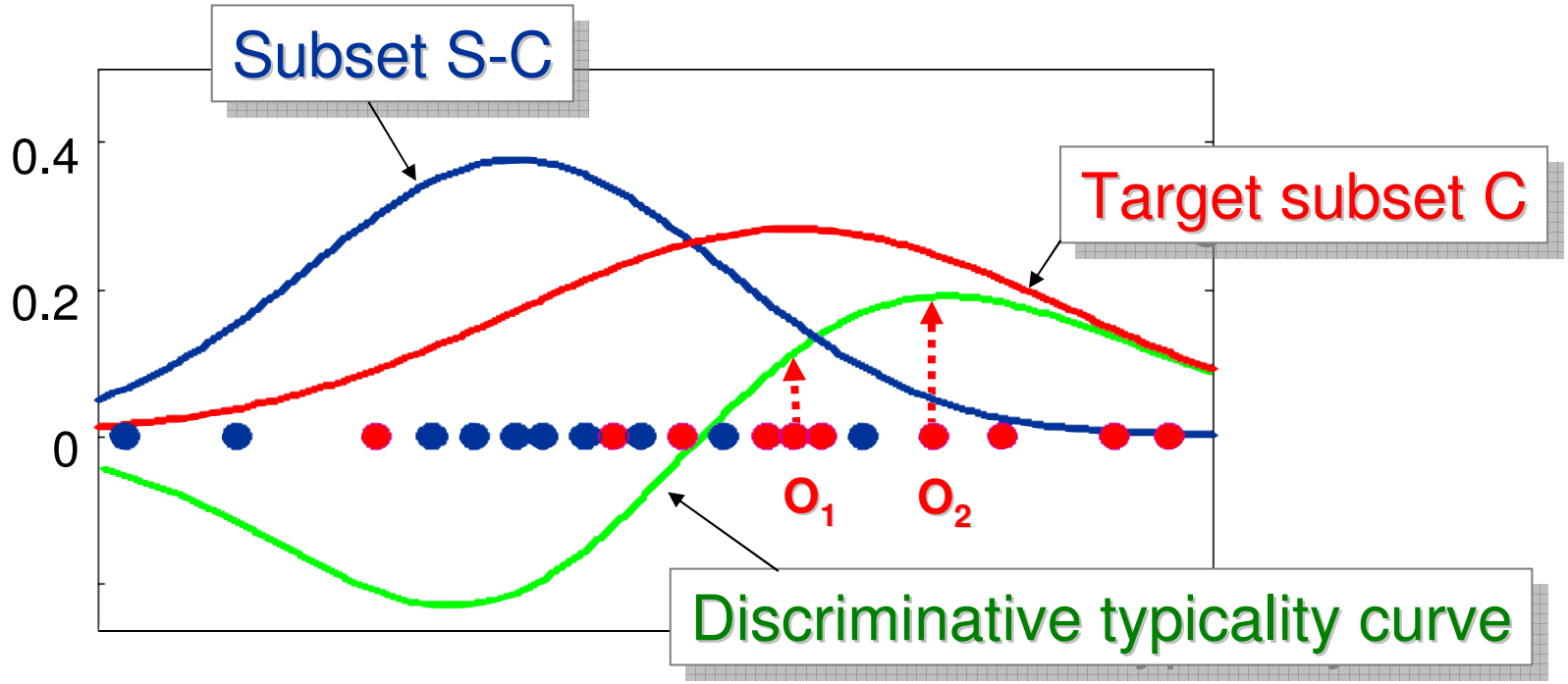
$$DT(o,C,S) = f(o \mid C) f(C) - f(o \mid (S - C)) f(S - C)$$

The typicality of o in the target set C

The typicality of o in S-C

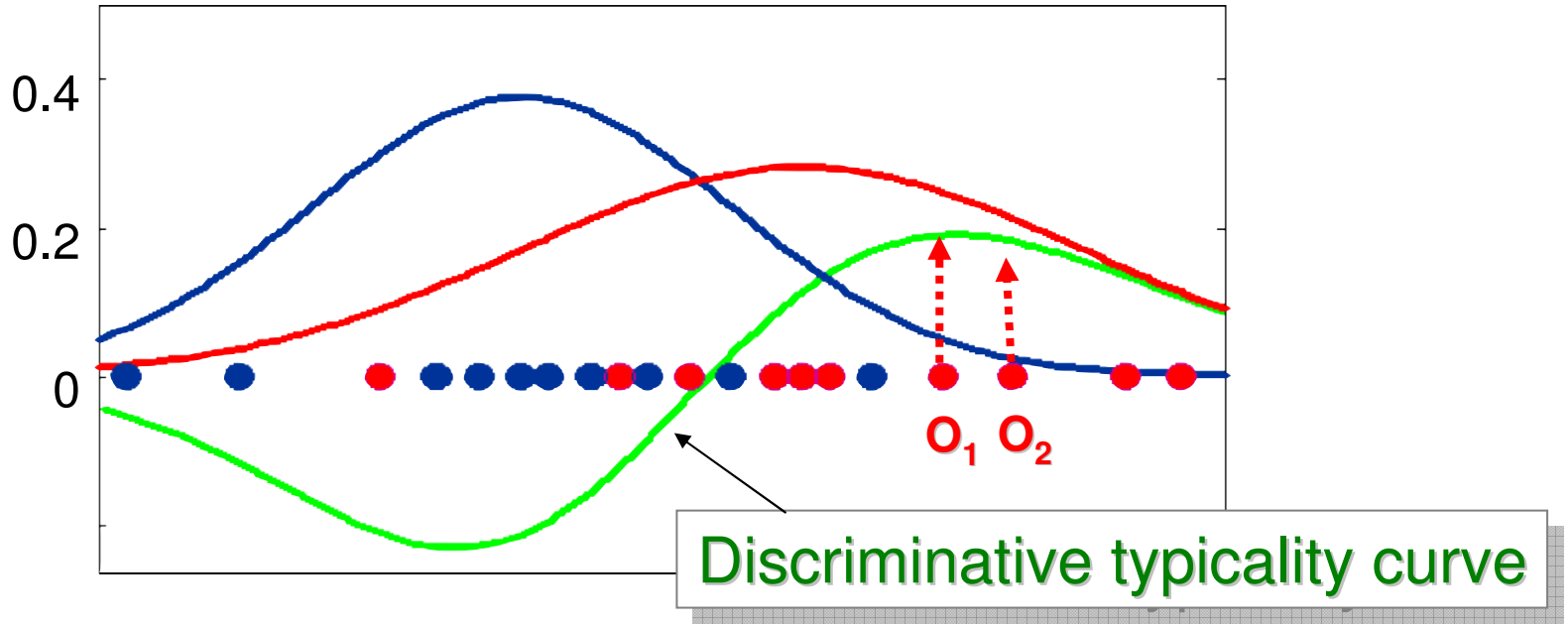
Discriminative Typicality

- The curve of discriminative typicality
  - O1 is more typical than O2
  - But O1 is not as discriminative as O2



Top-k Discriminative Typicality

- A top-k discriminative typicality query finds the object O in target subset C with the largest discriminative typicality



Summary: Typicality Definitions

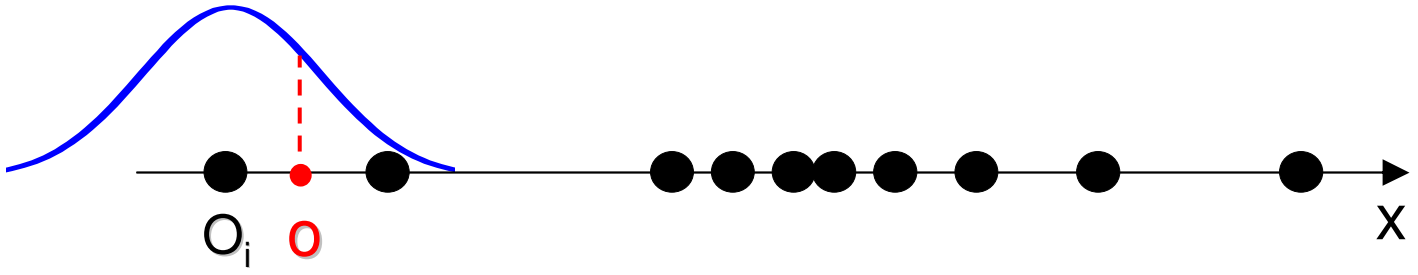
- Simple typicality
  - The *membership probability of o in S*
- Discriminative typicality
  - the *difference between its membership probability in the target set C and S-C*
- Challenges of Query Evaluation
  - How to compute the probability density of an object?
  - How to find the objects with the highest probability density?

Probability Density Estimation

- Kernel Density Estimation
  - Gaussian kernel

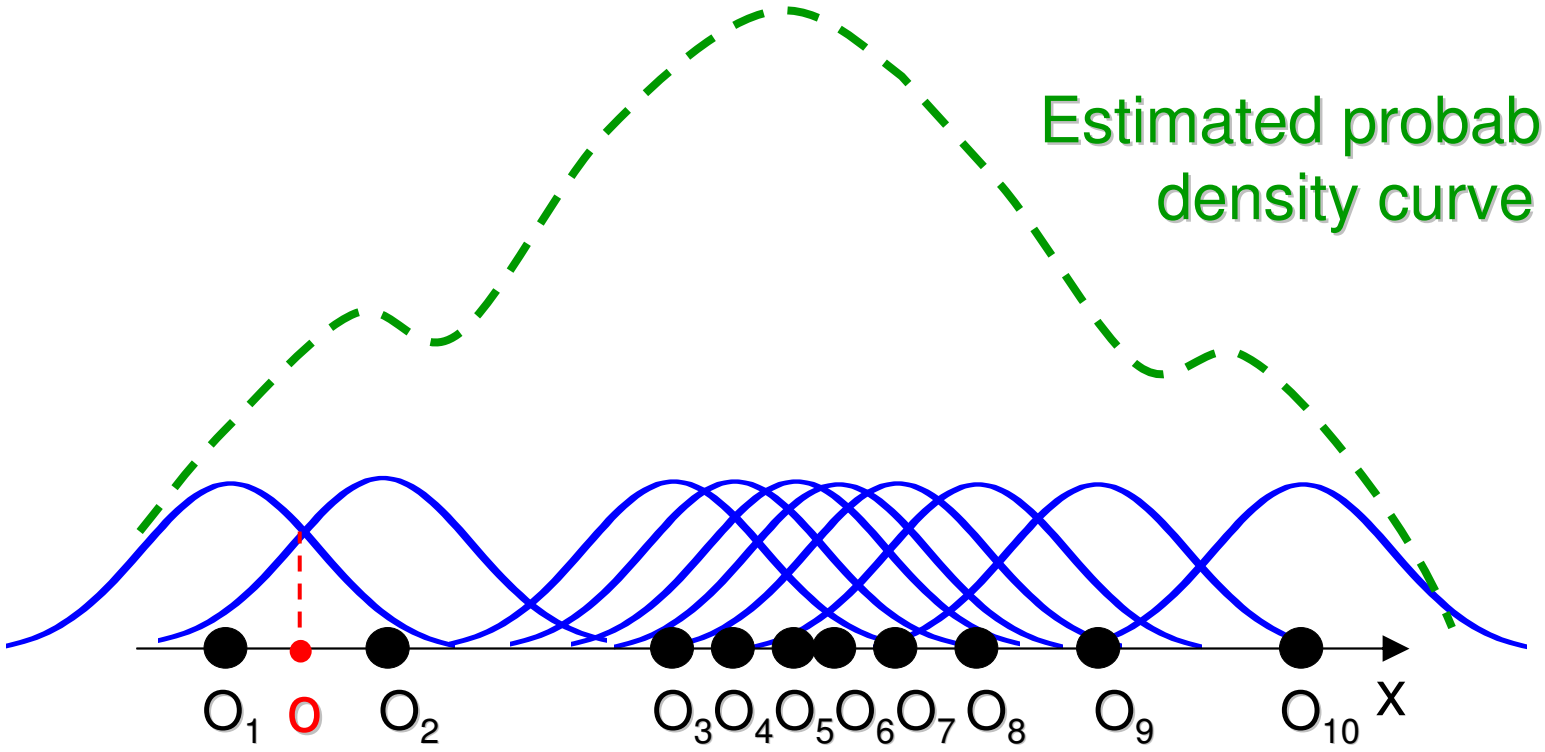
$$G_h(o, O_i) = \frac{1}{\sqrt{2\pi}} e^{-\frac{dist(o,O_i)^2}{2h^2}}$$

Gaussian kernel function curve



Kernel Density Estimation

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} G_h(o, O_i) = \frac{1}{n\sqrt{2\pi}}\sum_{i=1}^{n} e^{-\frac{dist(o,O_i)^2}{2h^2}}$$

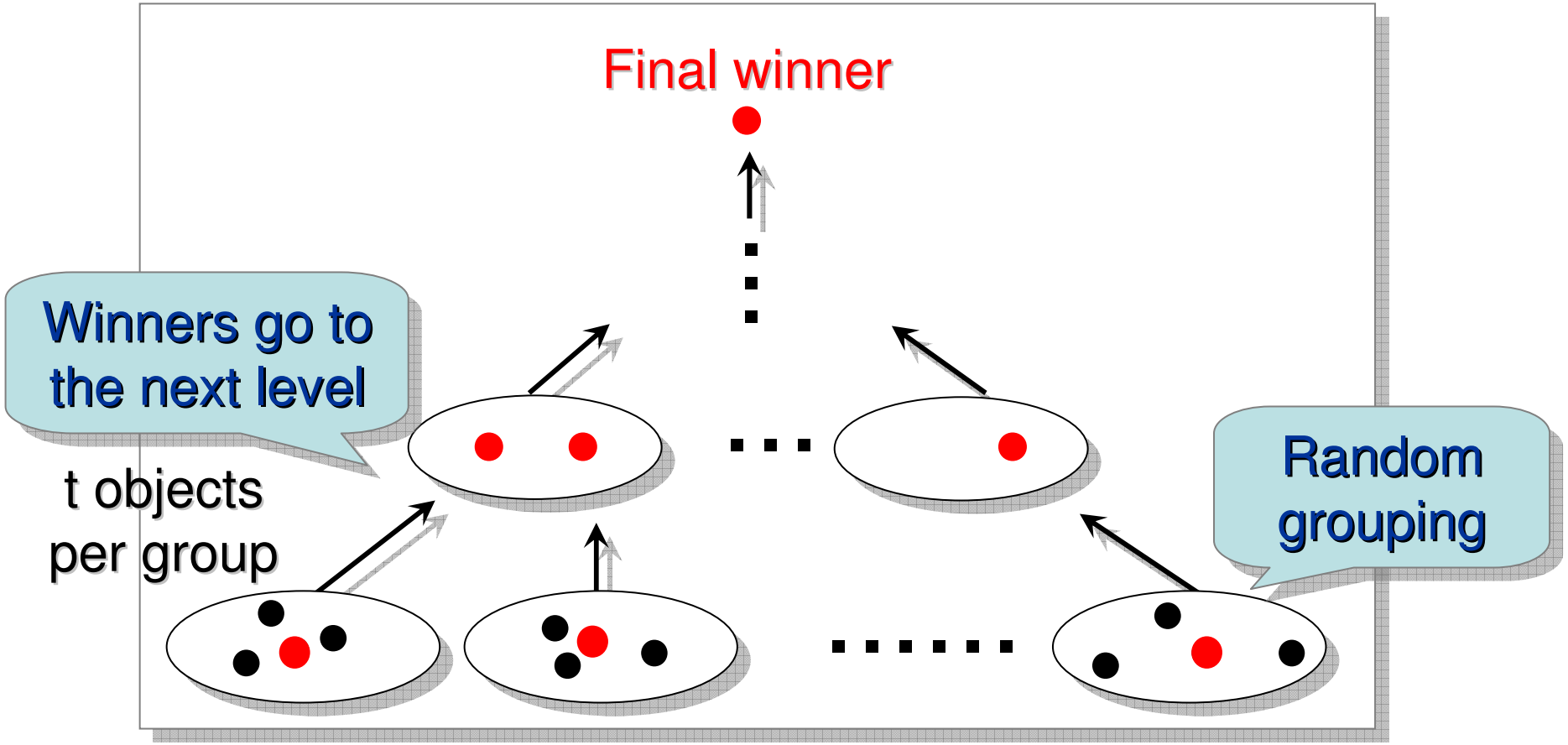


An Exact Algorithm

- Framework
  - For each object O, compute its simple typicality score (i.e., probability density at O).
  - Return the k objects with the highest typicality scores.
- Complexity
  - $O(n^2)$, where n is the number of objects in the data set.
  - Complexity similar to the discrete 1-median problem.

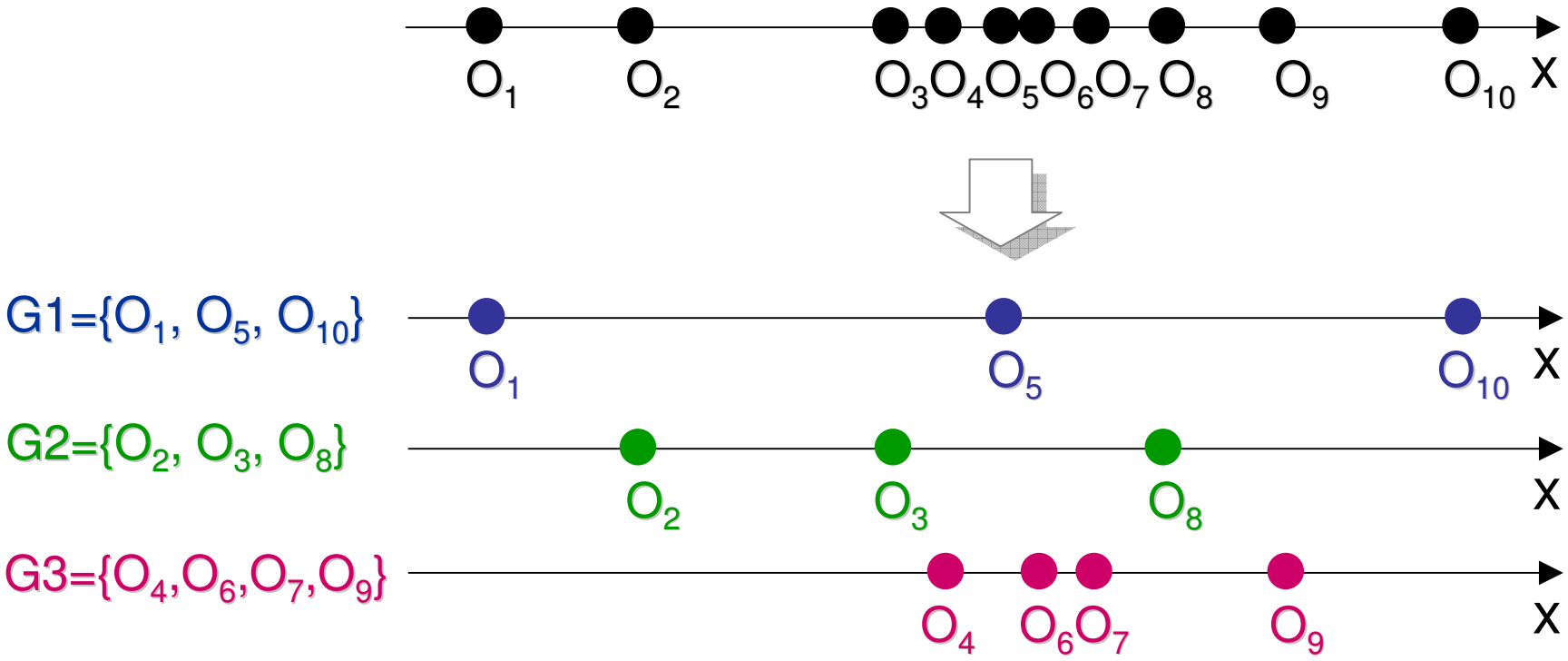
RT: Randomized Tournament

- The typical objects in a random sample are very likely to be typical in the data set



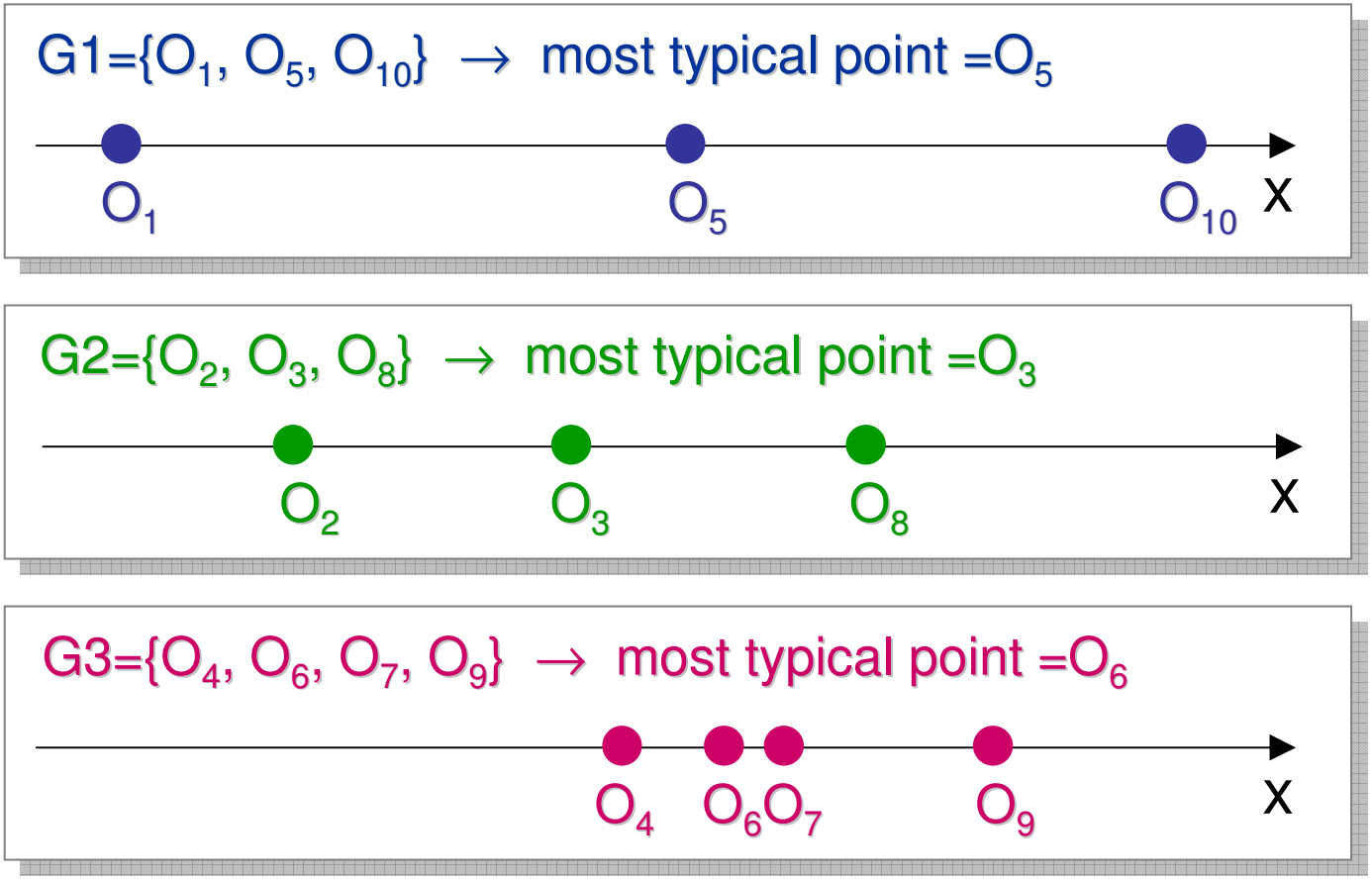
RT: Randomized Tournament

- *Step 1*: Given a group size t, randomly partition the data set into $\lceil n/t \rceil$ groups



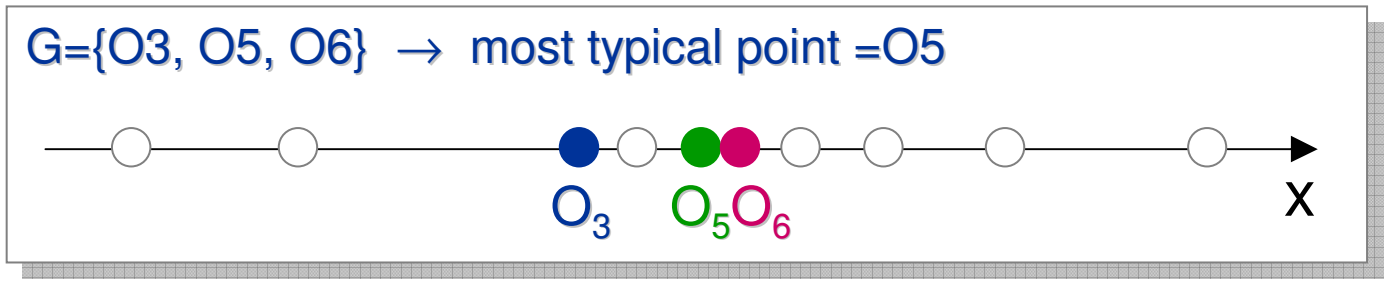
RT: Randomized Tournament

- *Step 2*: Find the most typical point in each group



RT: Randomized Tournament

- *Step 3*: Partition the winners again and repeat the tournament, until the final winner is found.



The approximate most typical point is O_5

RT: Randomized Tournament

- Complexity: $O(tn) = O(t^2) \times O(n/t)$
  - Complexity in each group: $O(t^2)$
  - Total number of groups: $O(n/t)$

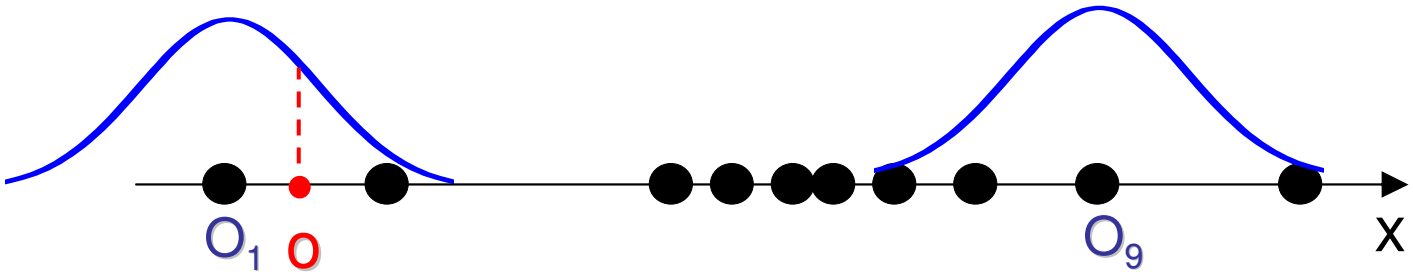
- Analysis
  - Pros: very efficient.
  - Cons: no approximation quality guarantee.

Can we find an approximation algorithm with quality guarantee?

Local Typicality Approximation

- Observation in Kernel Density Estimation
  - Typicality of o is the sum of contribution from other objects in the data set.
  - The contribution decays exponentially as the distance to o increases.

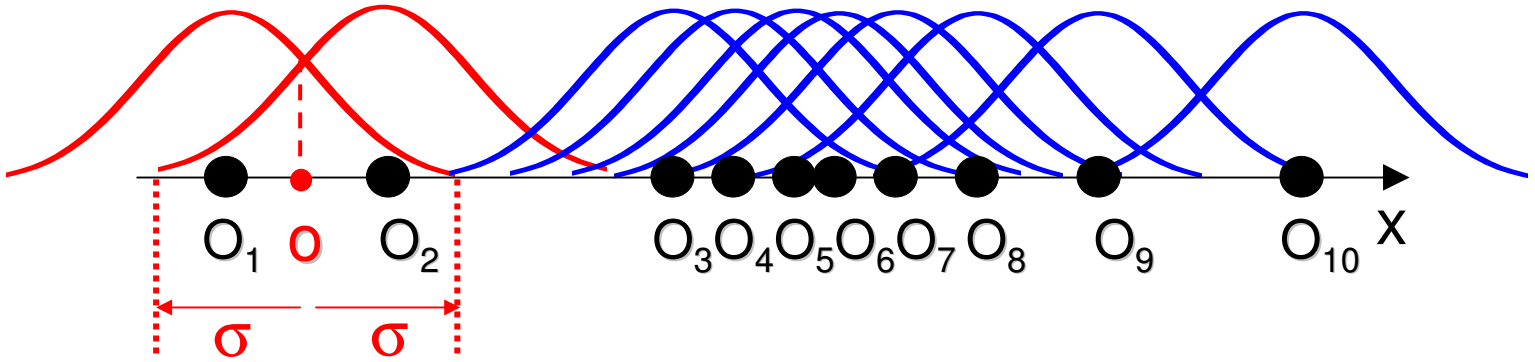
Gaussian kernel function curve



- Local Approximation
  - To approximate the typicality score of O, we only need to consider the points in the *neighborhood* of O.

Local Typicality Approximation

- σ-local neighborhood
  - The σ-local neighborhood of an object o contains the points whose distance to o is at most σ.



- Local simple typicality
  - The local simple typicality of a point o is the probability density estimated by its σ-local neighborhood.

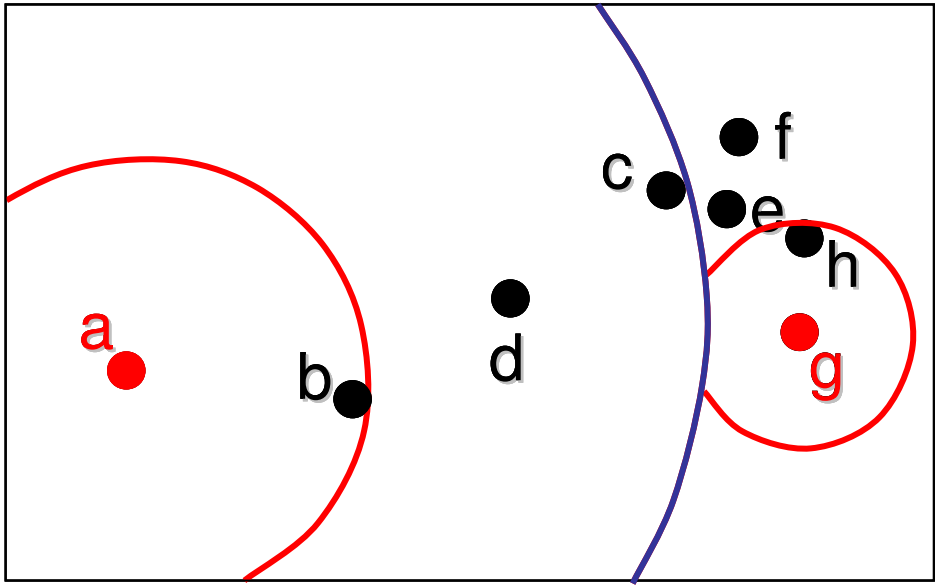
DLTA

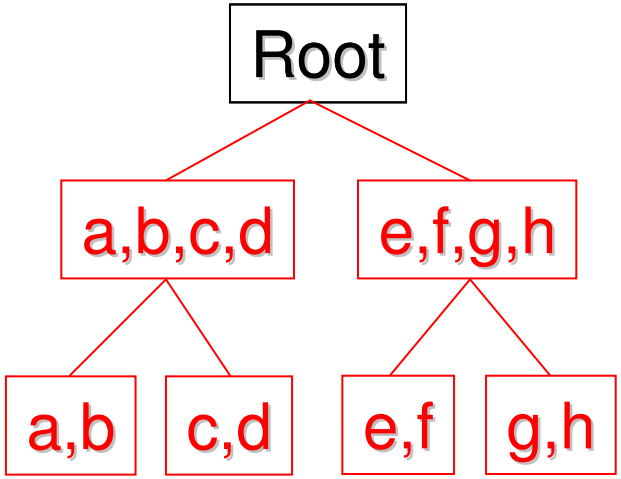
- Direct Local Typicality Approximation
  - For each object O, compute its *local simple typicality*.
  - Return the k objects with the greatest local simple typicality scores.
- Approximation Quality
  - Suppose O is the most typical object, and O' is the object with largest local typicality. Then

$$T(O,S) - T(O',S) \leq \frac{1}{\sqrt{2\pi}} e^{-\frac{\sigma^2}{2h^2}}$$

Computing σ-Local Neighborhood

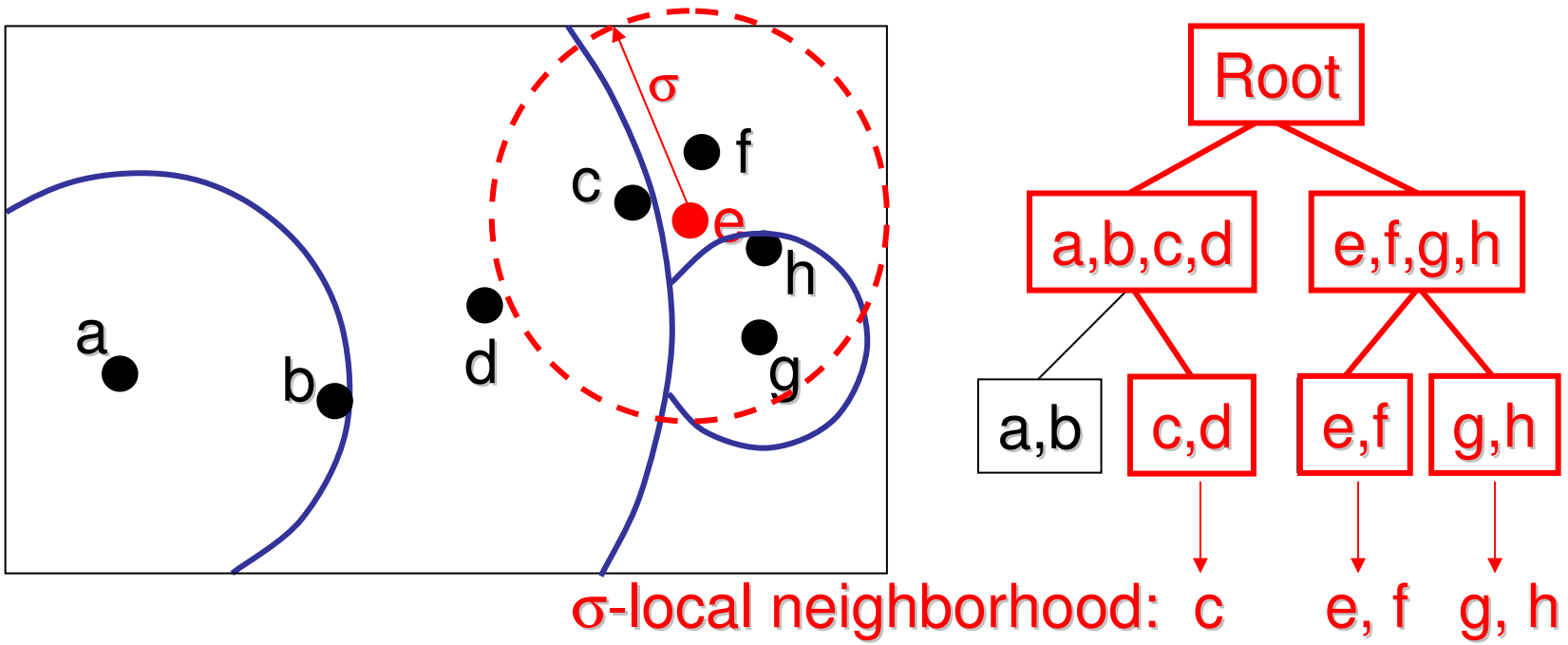
- VP-Tree
  - An index structure in generic metric space.
  - Support various similarity search.





Computing σ-Local Neighborhood

- Computing σ-local neighborhood
  - The σ-local neighborhood of a point p can be computed by recursive tree search

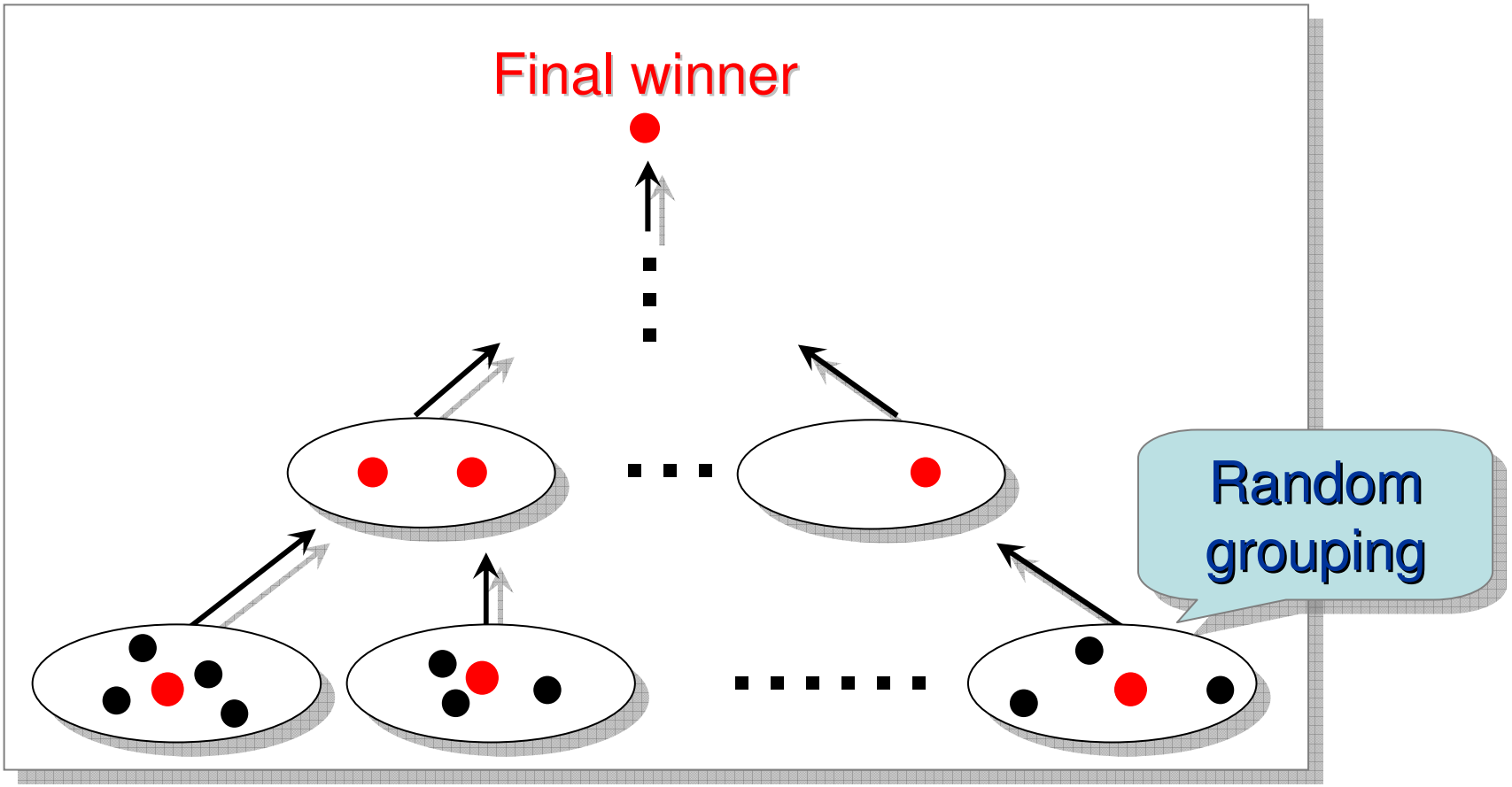


DLTA: Summary

- Computing the local typicality of o
  - O(LN(o, σ) time.
    - LN(o, σ) is the σ-local neighborhood of o
    - The σ-local neighborhood may contains all the points
- Complexity: $O(n^2)$
- Analysis
  - Pros: provide constant-factor approximation; efficient in practice.
  - Cons: time complexity is still $O(n^2)$.

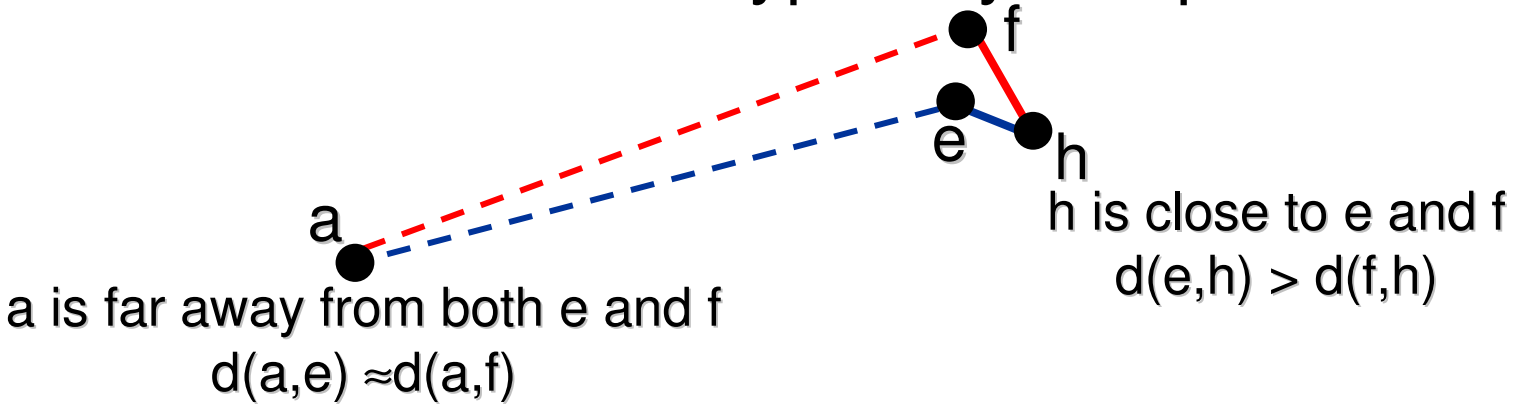
LT3: Local Tournament

- Review: randomized tournament



LT3: Local Tournament

- Review: Local Typicality Approximation
  - Only the objects in the local neighborhood contributes a lot in typicality computation

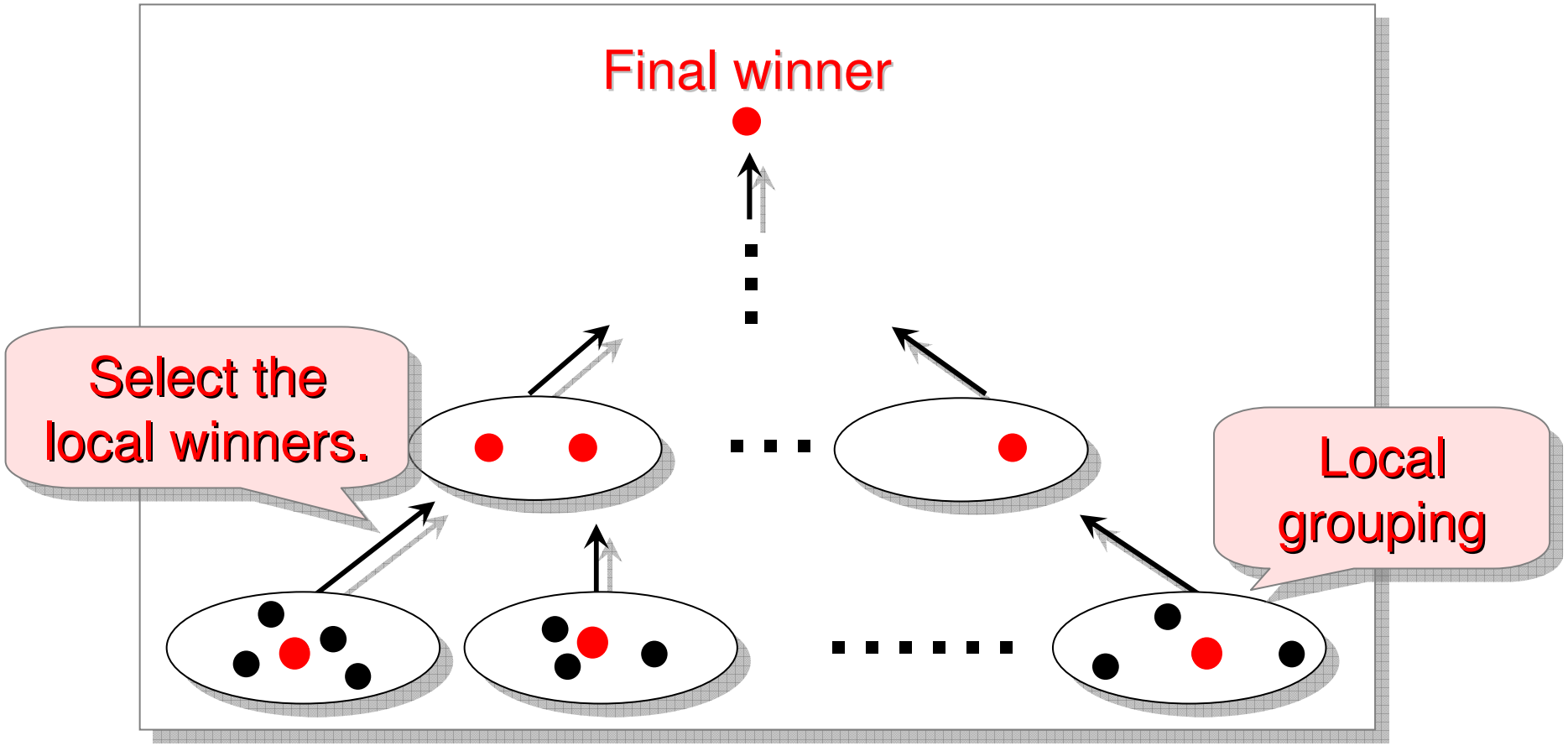


- Local Tournament

Tournament mechanism + Local typicality approximation → Local tournament

LT3: Local Tournament

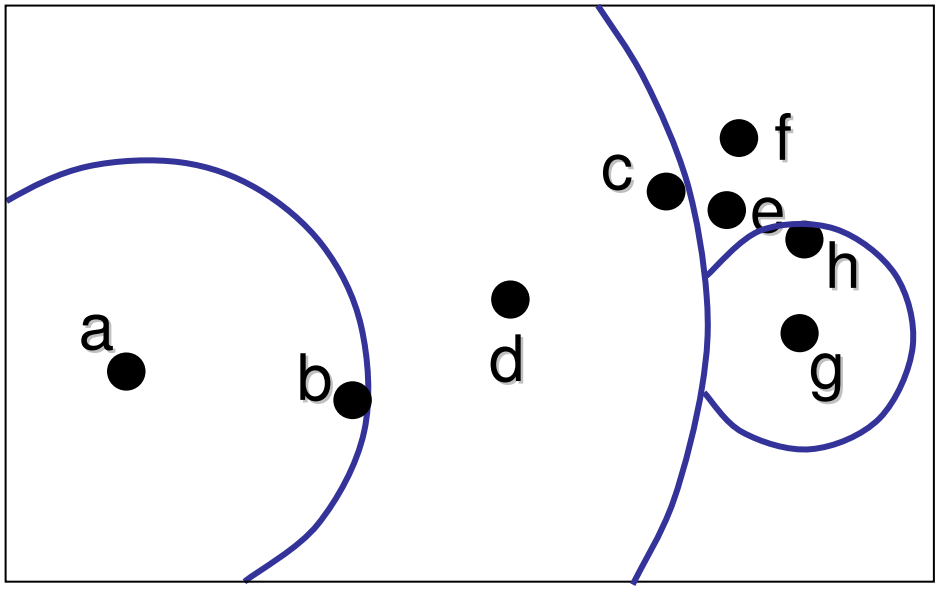
- Combined with local typicality approximation.

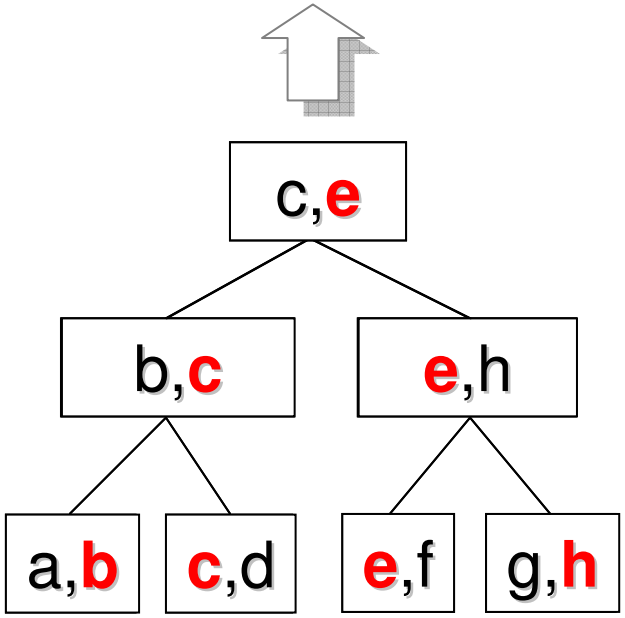


Local Tournament

- Computing the most typical point
  - Conduct tournaments from bottom up.

The approximate most typical point: e

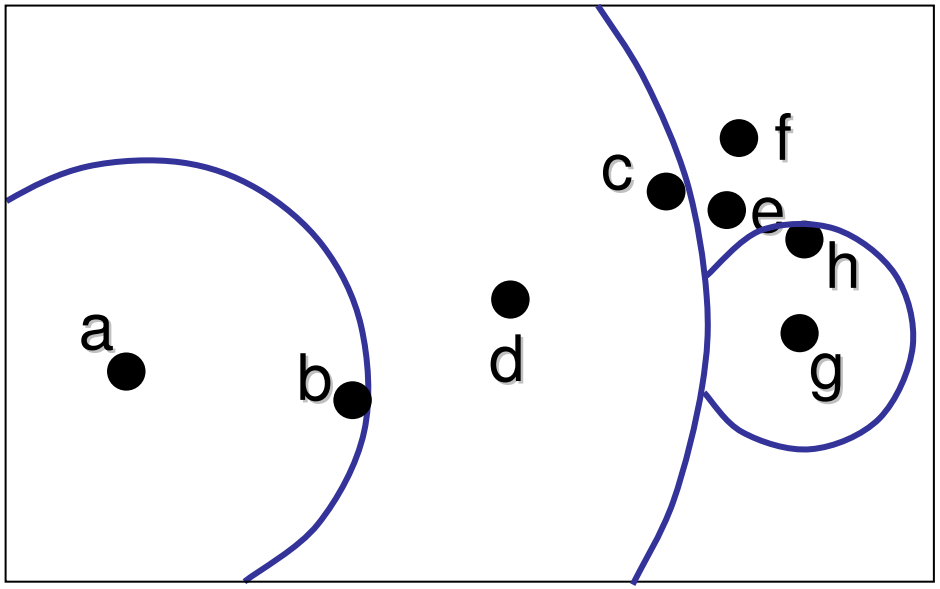


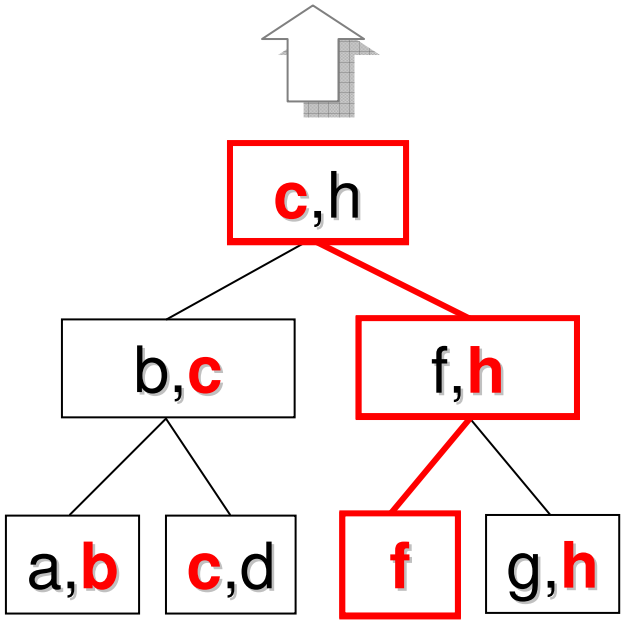


Local Tournament

- Computing the 2nd most typical point
  - Remove the most typical point e
  - Re-conduct containing e

The approximate 2nd most typical point: c





Local Typicality Computation by Sampling

- Sample the σ-local neighborhood of o
  - Draw a random sample S in the σ-local neighborhood of a node.
  - Estimate the local typicality using the random sample S.
- Chernoff-Hoeffding bound
  - (ε,δ)-approximation of local typicality if sample size:

$$|S| > \frac{3\sqrt{2\pi}\cdot e^{\frac{\sigma^2}{2h^2}}\cdot \ln\frac{2}{\delta}}{\varepsilon^2}$$

Analysis of LT3

- Summary
  - Local tournament, combining
    - Local typicality approximation.
    - Randomized tournament.
  - Local neighborhood sampling.
- Approximation quality
  - Probabilistic approximation quality guarantee
- Time complexity
  - O(nlogn)
    - VP-tree contruction

Algorithm Summary

Algorithm	Answer Quality	Time Complexity	Techniques
Exact Algorithm	Exact	$O(n^2)$	
RT	No quality guarantee	O(kn)	Randomized tournament
DLTA	Constant approximation ratio	$O(n^2)$	Local typicality approximation; VP-tree index
LT3	Probabilistic quality guarantee	O(nlogn)	Local typicality approximation; Local tournament; Uniform sampling

Empirical Study

- Real Data Sets
  - Zoo Database from the UCI Machine Learning Database Repository.
  - NBA 2005-2006 Season Statistics from Yahoo! Sports.
- Synthetic Data Sets
  - Quadraped Animal Data Generator from the UCI Machine Learning Database Repository.

Zoo Database

- 40 mammals, 20 birds, 14 fish, 10 invertebrate, 8 insects, 5 reptiles, 3 amphibians

Category	Most typical	Most discriminative typical	Most atypical
Mammal	Boar, Cheetah, Leopard, Lion, Lynx, Mongoose, Polecat, Puma, Raccoon, Wolf	Boar, Cheetah, Leopard, Lion Lynx, Mongoose, Polecat, Puma, Raccoon, Wolf	Platypus
Bird	Lark, Pheasant, Sparrow, Wren	Lark, Pheasant, Sparrow, Wren	Penguin
Fish	Bass, Catfish, Chub, Herring, Piranha	Bass, Catfish, Chub, Herring, Piranha	Carp
Invertebrate	Crayfish, Lobster	Crayfish, lobster	Scorpion
Insect	Moth, Housefly	Gnat	Honeybee
Reptile	Slowworm	Pitviper	Seasnake
Amphibian	Frog	Frog	Newt, Toad

NBA 2005-2006 Season Statistics

The top-3 most typical players

Name	Position	Minutes	PPG	3PT	Rebounds	Ast	Blk	PF
Danny Granger	Forwards	22.6	7.5	1.6	4.9	1.2	0.8	2.7
Devean George	Forwards	21.7	6.3	3	3.9	1.0	0.5	2.2
Michael Finley	Guards	26.5	10.1	5	3.2	1.5	0.1	1.3

The top-3 most atypical players

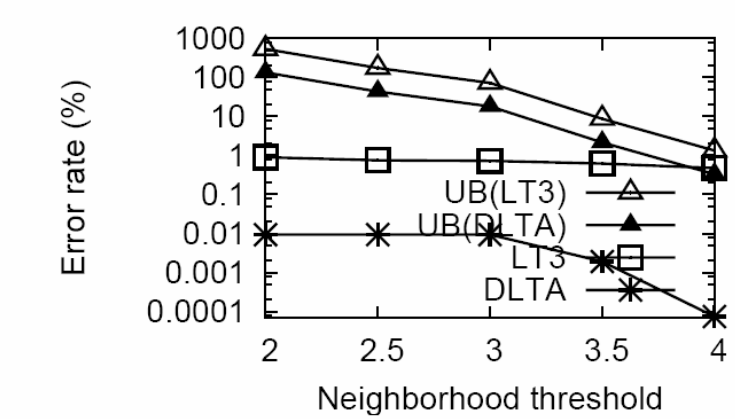
Name	Position	Minutes	PPG	3PT	Rebounds	Ast	Blk	PF
Tracy McGrady	Guards	37.1	24.4	6.6	6.6	4.8	0.9	1.9
Allen Iverson	Guards	43.1	33.0	4.1	3.2	7.4	0.1	1.7
Doug Christie	Guards	26.5	3.7	0.1	1.9	2.0	0.1	1.4

Approximation Quality

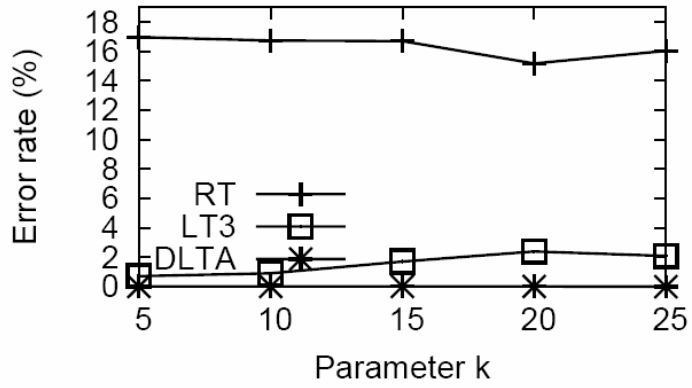
- DLTA has the best approximation quality.
- LT3 performs a little worse than DLTA, but the overall error rate is still under 5%.
- The error rate of RT is around 10% to 15%.

Approximation Quality
DLTA > LT3 > RT

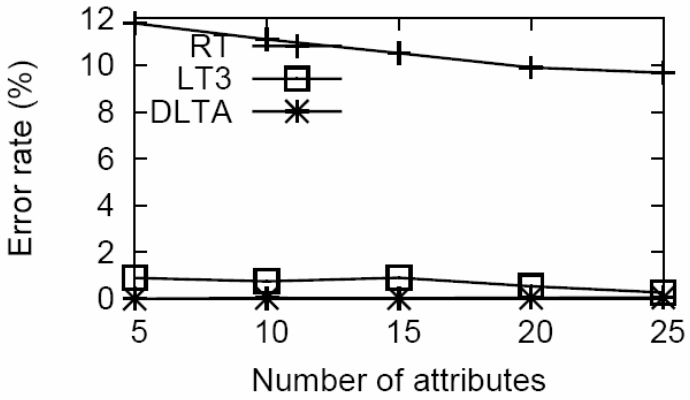
Approximation Quality



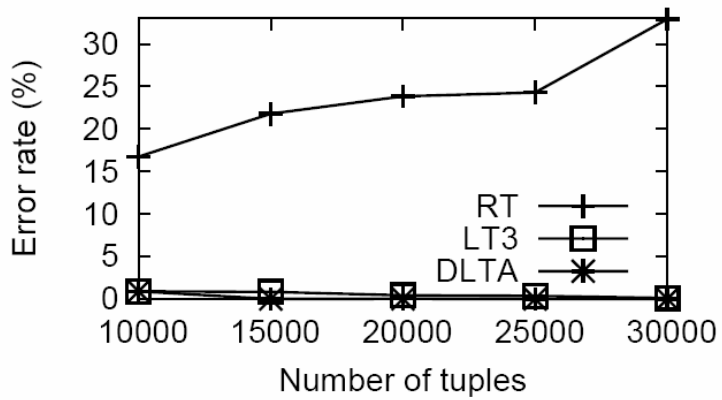
(a) Error rate vs. neighborhood.



(b) Error rate vs. k.



(c) Error rate vs. dimensionality.



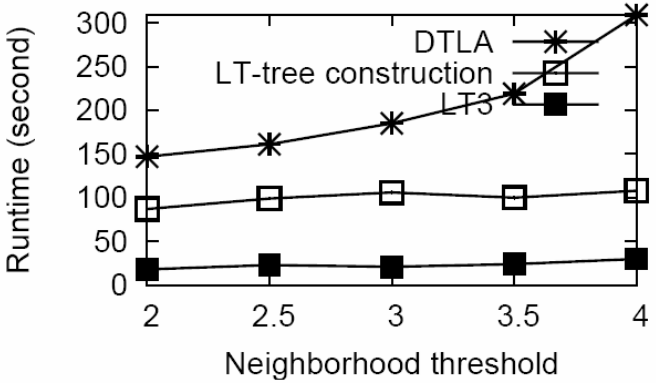
(d) Error rate vs. cardinality.

Efficiency and Scalability

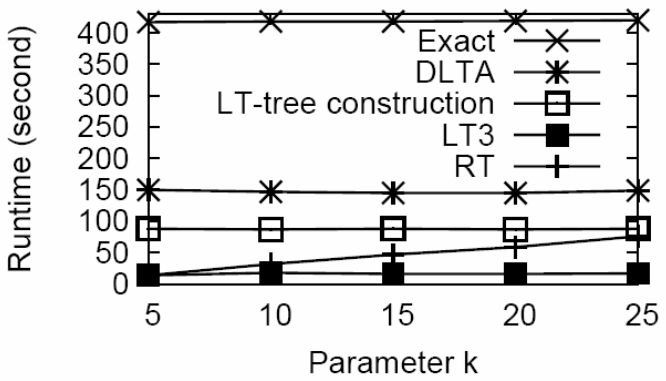
- All three algorithms are scalable.
  - 100,000 tuples, 25 dimensions.
- RT has a linear scalability.
- LT3 has a better performance and scalability than DLTA on large databases.
- DLTA is not as efficient as RT and LT3, but still a lot better than the exact algorithm.

Efficiency and scalability:
RT > LT3 > DLTA

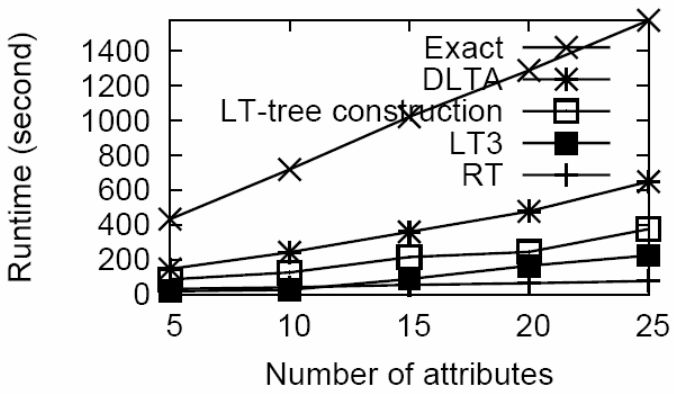
Efficiency and Scalability



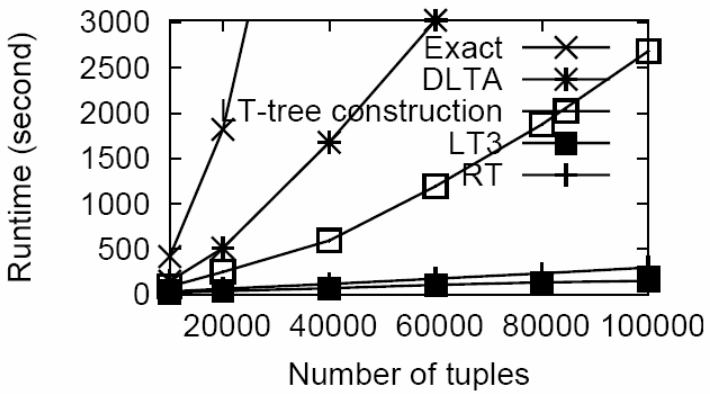
(a) Runtime vs. neighborhood.



(b) Runtime vs. k.



(c) Runtime vs. dimensionality.



(d) Runtime vs. cardinality.

Related Work

- Psychology and Cognitive Science
  - No efficient query answering algorithm.
- Top-k Ranking Queries
  - Requires a scoring function defined by users.
- Discrete 1-Median Problem
  - Different optimization functions.
- Spatially-decaying aggregation
  - Defined on Euclidian plane or graphs.

Summary

- Top-k typicality queries
  - Simple typicality and discriminative typicality.
  - Applications: summarization of answer set, etc.
- A series of efficient algorithms
  - Exact algorithm.
  - Randomized tournament.
  - Local typicality approximation: DLTA and LT3.
- Empirical study
  - Real data sets and synthetic data sets.
- Future direction
  - How to use typicality in data summarization?
  - How to evaluate the typicality of a group of objects?

Q & A

Thank you!