Query Routing Mechanisms in Self-organizing Search Systems

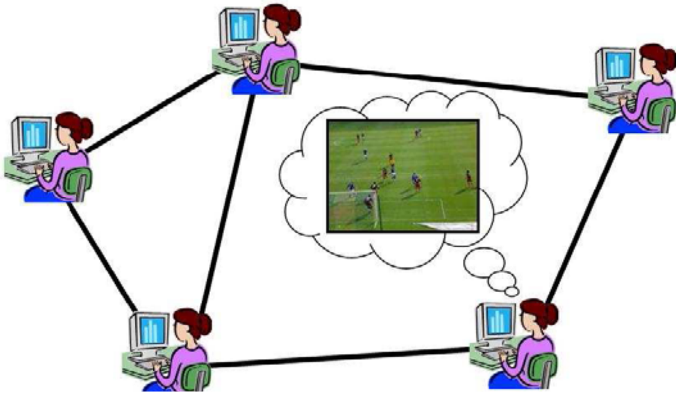
Vlastislav Dohnal, Jan Sedmidubský

Outline

- Self-organizing search systems
- Routing algorithm
- Confusability of queries
- Experimental trials
- Conclusions and future work

Self-organizing Search Systems

- A set of interacting components creating a desired outcome
  - Evolves in time and space
  - Inspired in sociology, biology
- Goal: search for information
- Properties:
  - Scalability
  - Adaptability
  - Robustness

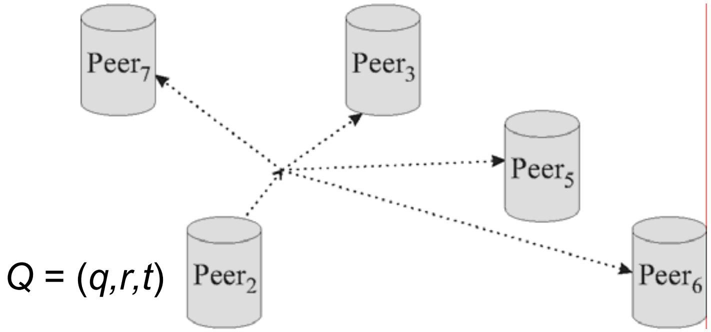


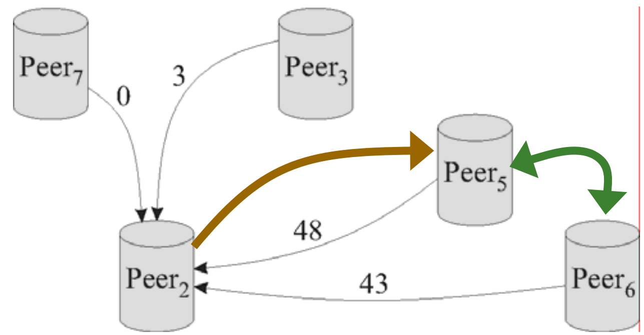
Metric Semantic Overlay

- Self-organizing system over a P2P network
- Metric space as data model
- Structure:
  - Peers
    - Data stored in the corresponding peer of underlying P2P network
    - Query history, list of exploration peers
  - Relationships
    - Exploited for query routing
    - Created by analyzing queries and their answers

Relationships

- Created according to peers' answers to the processed query

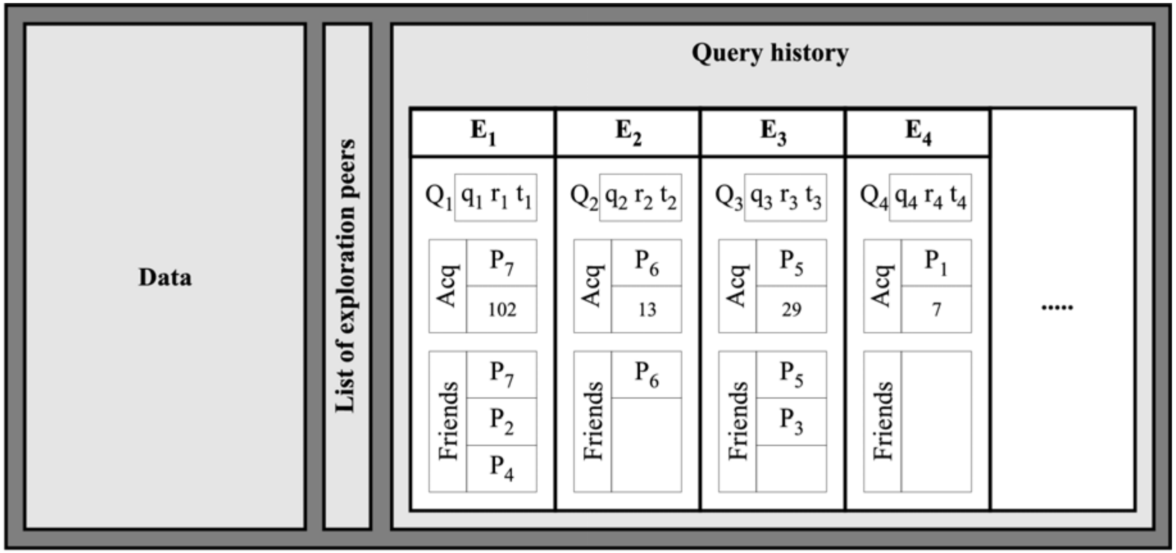




- Acquaintance
  - Peer returning the biggest part of the answer
  - $Acq(Q_3) = Peer_5$
  - Acquaintance relationship: between $Peer_2$ and $Peer_5$
- Friends
  - Peers returning the significantly-large part of the answer
  - $Friends(Q_3) = \{Peer_5, Peer_6\}$
  - Friend relationships: between each pair of friends

Peer

- Query history
  - List of entries $E_1, \ldots, E_n$ representing the relationships
  - Each entry contains metadata about a processed query:
    - Query object, radius, timestamp
    - Acquaintance
    - List of friends



Query Routing

- At each peer *P*, a query *Q*=(*q*,*r*,*t*) is evaluated:
  - Inspect all entries of query history and take ones *most relevant* to *Q*
  - Forward *Q* to the *acquaintances* of these entries
  - In case of few relevant entries, Q is forwarded to some *exploration peers*.

  - If there is no more relevant entry, do:
    - Evaluate *Q* on local data
    - Ask all friends to answer *Q*
    - Return all answers to P_{start}

Relevancy of Entries

- By means of *Confusability*
  - $conf(Q,Q_t) \rightarrow [0,1]$
  - It measures closeness and extent of queries.

  - Identical queries: $conf(Q,Q) = 1$
  - Queries Q_t having $conf(Q,Q_t) \geq ct_{high}$ are *highly relevant* to Q
  - Queries Q_t having $conf(Q,Q_t) < ct_{low}$ are *irrelevant* to Q

  - Parameters: $ct_{low} = 0.3$ $ct_{high} = 0.8$

Measures of Confusability

- A new range query $Q=(q,r,t)$, a template query $Q_t=(q_t,r_t,t_t)$
  - $conf(Q,Q_t) \to [0,1]$
- Exponential function
  - B is constant, depends on data:
    - B = 1 / most frequent distance d
- Adaptive exponential function
  - Adapts to varying radii
- Adaptive Gaussian-like function
  - Overlapping queries are very similar
    - $d(q,q_t) \le r$

$$Exp(Q,Q_t) = e^{-B\,d(q,q_t)}$$

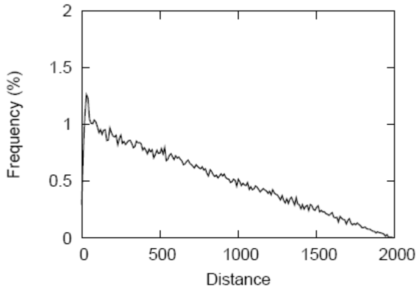
$$aExp(Q,Q_t) = e^{-\frac{\ln ct_{low}}{-r-r_t} d(q,q_t)}$$

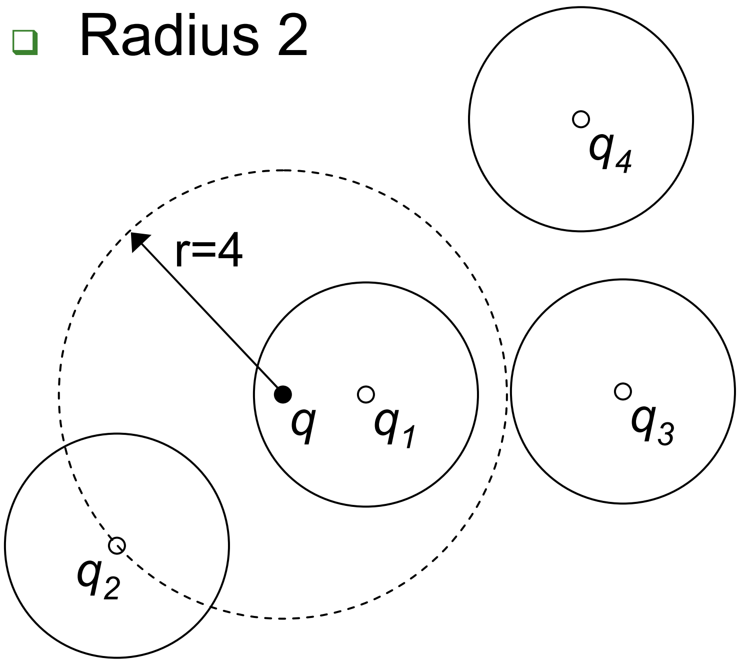
$$aGauss(Q,Q_t) = e^{-B\,d(q,q_t)^C}$$

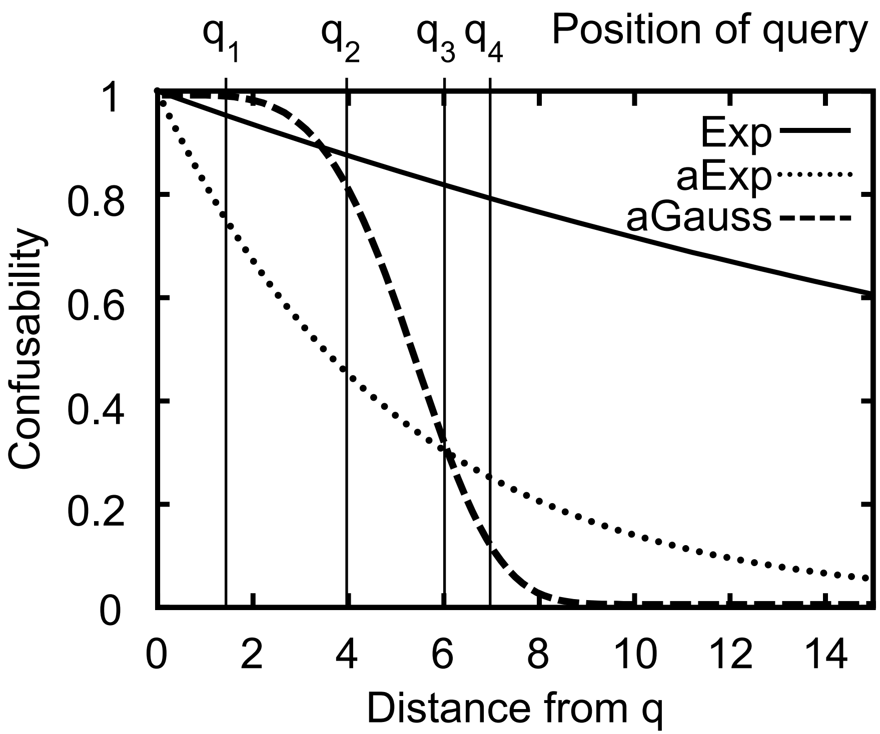
$$B = \frac{\ln ct_{low}}{(-r-r_t)^C} \qquad C = \frac{\ln \frac{\ln ct_{high}}{\ln ct_{low}}}{\ln \frac{r}{r+r_t}}$$

Measures of Confusability – Example

- 2-d data, uniform distr., Euclidean dist.
  - Most-frequent distance 30.0
  - Exp function: B=1/30
  - Radius 2

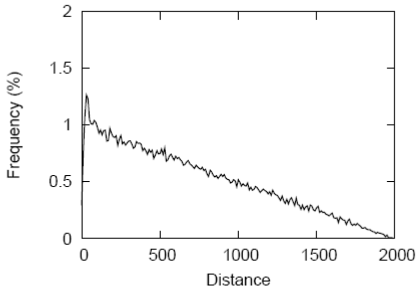




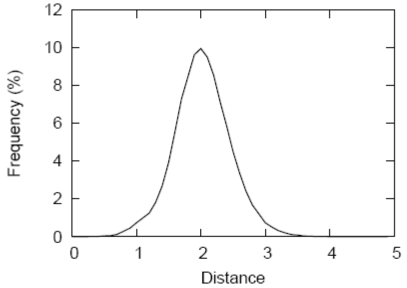


Experimental Comparison

- Synthetic dataset – 100,000 2-d vectors
  - [0;1,999] x [0;49] space
  - Each peer contains 50 objects having the same *x*-coordinate

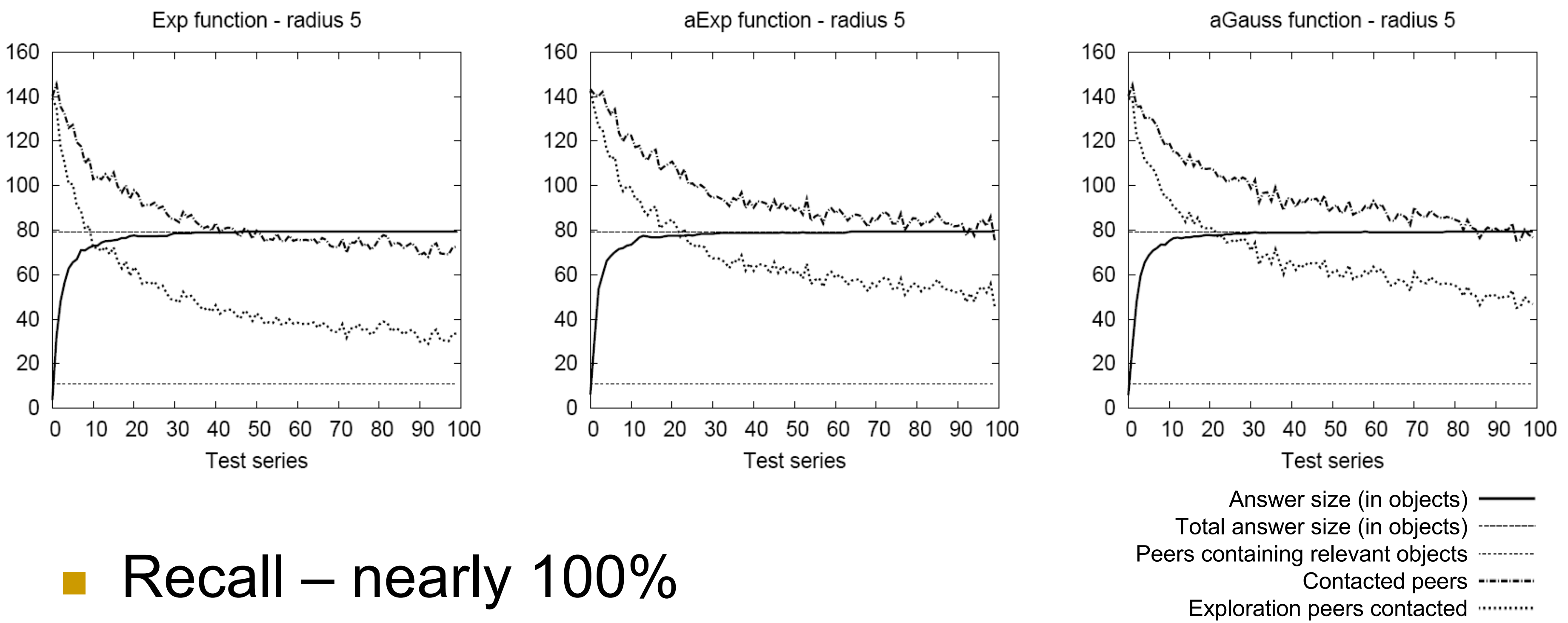


- Real-life dataset – 100,000 image features
  - Subset of CoPhIR dataset
  - Each peer contains 50 objects following M-Chord data-distribution principles



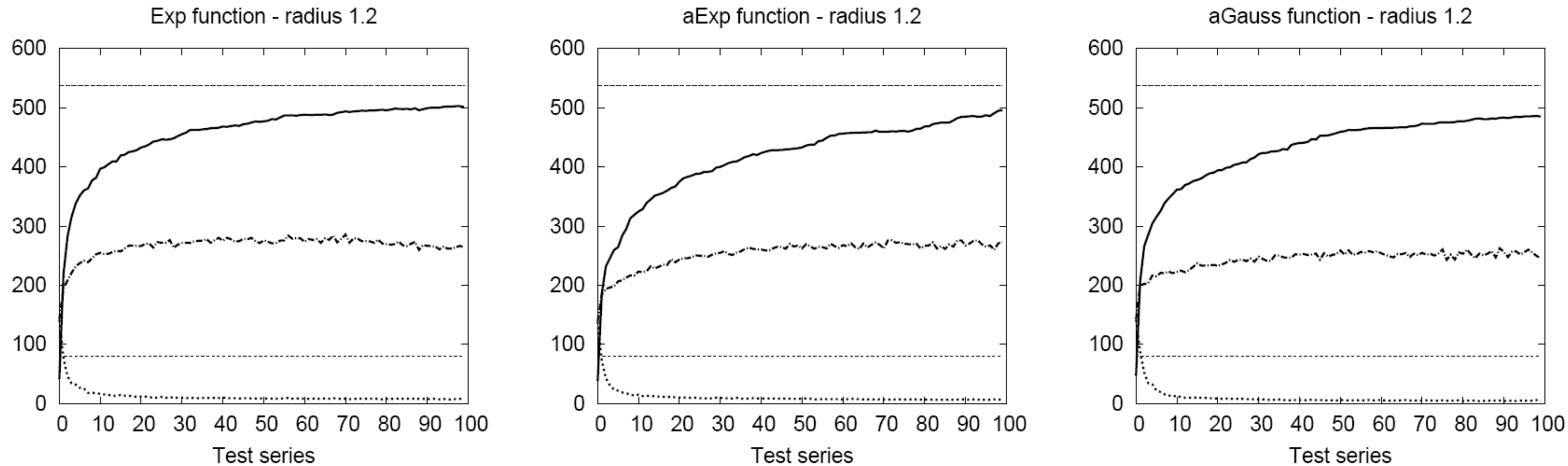
- List of exploration peers is initialized to just 50 random peers.
- Repeating the batch:
  - Training queries – 50 random objects, varying radii
  - Testing queries – 5 objects, same radius

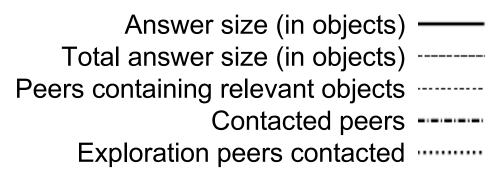
Experiment Results – 2-d, rad=5



- Recall – nearly 100%
- Costs – increased for *aExp* and *aGauss*
  - These functions are below *Exp*, so more exploration peers are used.

Experiment Results – CoPhIR, rad=1.2





- Recall – nearly 90%
- Costs – almost identical
  - Distance to the nearest neighbor is quite large, so *Exp* returns low values too. ⇒ The same number of exploration peers.

Conclusions

- Contribution
  - Adaptive functions focus more on similar queries (overlapping)
  - Adaptive functions are data independent.
  - Navigation is more focused
    - Contacting fewer peers that are promising to contain data
- Future work
  - Advanced filtering techniques to decrease costs
  - Detecting when the system is adapted (learned)
  - Management of query history