Partner MU-Brno
WP5 – complex queries

Pavel Zezula
Faculty of Informatics
Masaryk University, Brno

Complex Queries

- Several different attributes (features) of every object
  - similarity measure for every attribute is different
- Complex queries
  - multiple simple similarity queries on attributes
  - ranks combined by an aggregation function

- Find the best matches of *circular* **shape** objects with *red* **color**
  - the best match for circular shape or red color needs not be the best match combined!!!

Complex Queries – Definition

- Assume that object $o \in D$ has *m* attributes $(o_1, o_2, \ldots, o_m)$
  - every attribute is comparable by a distance function d_i
  - the value of an aggregation function *t*

    $$t(d_1(q_1, o_1), d_2(q_2, o_2), \ldots, d_m(q_m, o_m))$$

    represents the “score” of object *o* with respect to query object *q*
  - function *t* must be monotonous
  - normalized similarity grades can be used instead of distances

    $$x_i \in [0;1], i = 1, \ldots, m$$

    - represents how similar is the object *o* to query *q* in respective attribute

The $\mathcal{A}_0$ Algorithm

- Retrieve k top objects with respect to $q = (q_1, q_2, \ldots, q_m)$
- For each attribute i
  - objects delivered in decreasing similarity to q_i
  - incrementally build sets X_i with best matches till

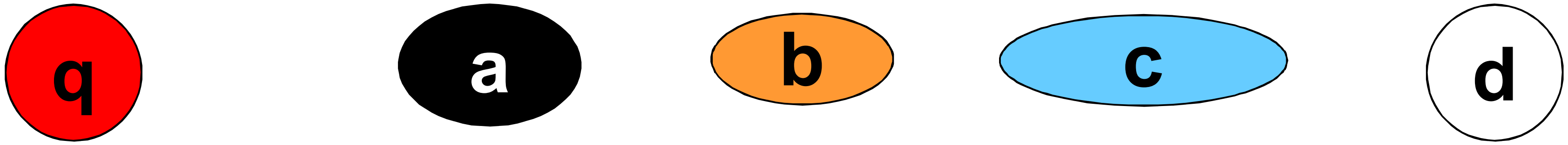
$$\forall i \,|\cap_i X_i| = k$$

- For all $o \in \cup_i X_i$
  - compute function ***t*(*o*)** – needs more dist. comp.
  - sort results according to values of ***t*(*o*)**
  - return k first objects

Threshold Algorithm TA

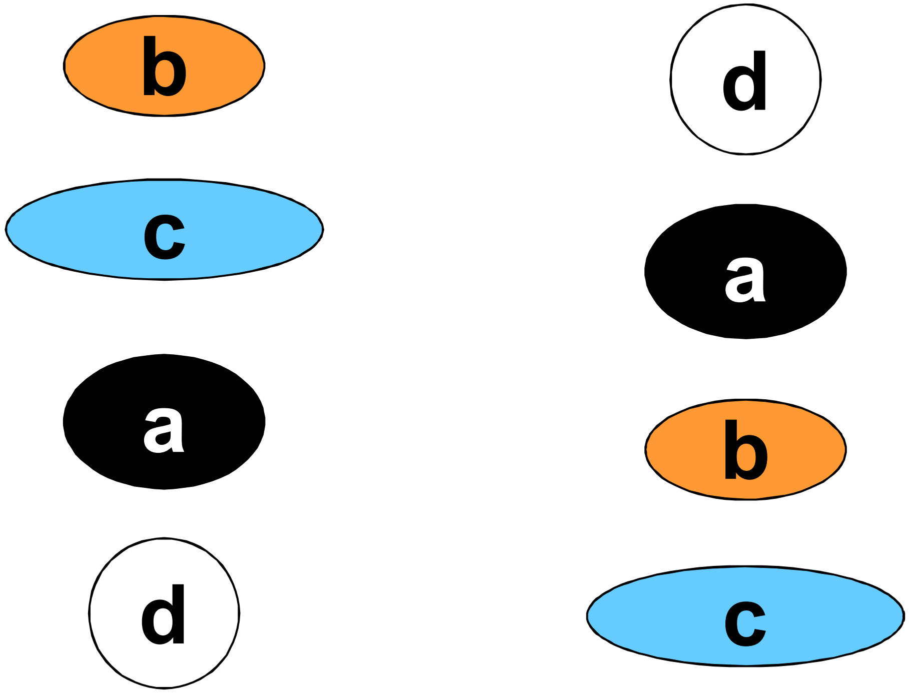
- Retrieve *k* top objects with respect to $q = (q_1, q_2, \ldots, q_m)$
- Incrementally retrieve objects in every attribute *i*
  - objects in decreasing similarity stored in lists X_i
- Let μ_i be the maximal grade (distance) seen in list X_i
- The **threshold value** is defined as $t(\mu_1, \mu_2, \ldots, \mu_m)$
- For every object *o* retrieved in any list X_i
  - compute the score $t(o)$
  - if the score belongs to the best *k* scores seen so far
    - remember *o* and $t(o)$ - only first *k* objects are stored
- Stop if at least *k* objects with scores up to the threshold are found

Example



$t(o) = avg(d_1,d_2)$: avg(3,2) = 2.5 avg(1,3) = 2 avg(2,4) = 3 avg(4,1) = 2.5

X_1 list (color) X_2 list (shape)



t(μ) = 1

t(μ) = 2

t(μ) = 3

t(μ) = 4

TA Properties

- Sorted access to objects for every attribute
  - index structure for every attribute
    - e.g. an index for color histograms and another one for shapes
  - incremental nearest neighbor search is needed
- Random access to objects
  - ability to compute score of object *o* in other attributes
    - e.g. for a particular object *o = (color, shape)*, find its color similarity $d(q_1,o_1)$ and shape similarity $d(q_2,o_2)$
- Special variants of threshold algorithm
  - restricting random accesses
  - restricting sorted accesses

Challenge for SAPIR

- Multilayer architecture of MESSI
- Expensive sorted access – incremental nearest neighbor is needed
- Efficient random access
- Minimization of the network communication costs is needed
- Inter- and intra-query parallelism tradeoff

Partner MU-Brno
WP7 – social networks

Pavel Zezula

Faculty of Informatics

Masaryk University, Brno