# SIMILARITY SEARCH
# The Metric Space Approach

Pavel Zezula, Giuseppe Amato,
Vlastislav Dohnal, Michal Batko

# Table of Contents

Part I: Metric searching in a nutshell

- Foundations of metric space searching
- Survey of exiting approaches

Part II: **Metric searching in large collections**

- Centralized index structures
- **Approximate similarity search**
- Parallel and distributed indexes

# Approximate similarity search

- Approximate similarity search overcomes problems of exact similarity search using traditional access methods
  - Moderate improvement of performance with respect to sequential scan
  - Dimensionality curse
- Similarity search returns mathematically precise result sets
  - Similarity is subjective so, in some cases, also approximate result sets satisfy the user
- Approximate similarity search processes query faster at the price of imprecision in the returned result sets
  - Useful for instance in interactive systems
    - Similarity search is an iterative process where temporary results are used to create a new query
- Improvements up to **two** orders of magnitude

# Approximate similarity search

- Approximation strategies
  - **Relaxed pruning conditions**
    - Data regions overlapping the query regions can be discarded depending on the specific strategy
  - **Early termination of the search algorithm**
    - Search algorithm might stop before all regions have been accessed

# Approximate Similarity Search

1. **relative error approximation (pruning condition)**
   - **Range and k-NN search queries**
2. good fraction approximation
3. small chance improvement approximation
4. proximity-based approximation
5. PAC nearest neighbor searching
6. performance trials

# Relative error approximation

- Let $o^N$ be the nearest neighbour of q. If

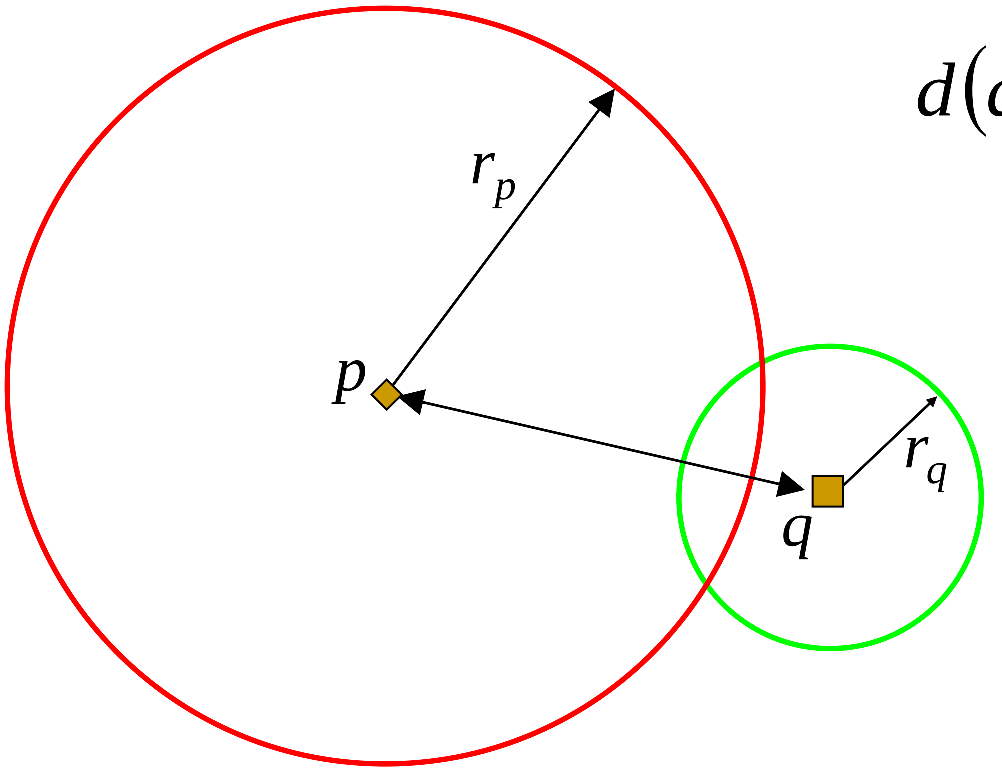
$$\frac{d\left(o^A, q\right)}{d\left(o^N, q\right)} \leq 1 + \varepsilon$$

  then $o^A$ is the *(1+ε)-approximate nearest neighbor* of *q*
- This can be generalized to the *k*-th nearest neighbor

$$\frac{d\left(o_k^A, q\right)}{d\left(o_k^N, q\right)} \leq 1 + \varepsilon$$

# Relative error approximation

- Exact pruning strategy:

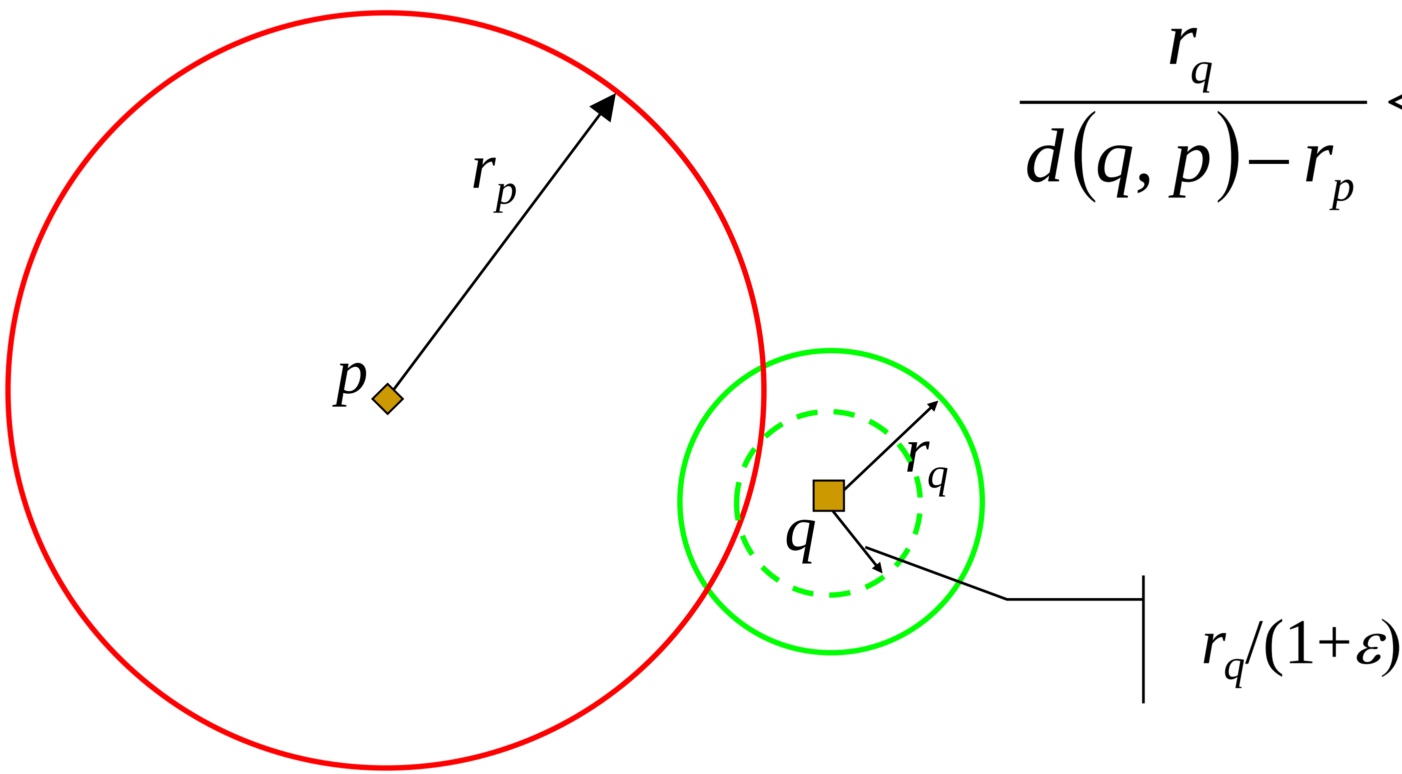



$$d(q, p) > r_q + r_p$$

# Relative error approximation

- Approximate pruning strategy:

$$\frac{r_q}{d(q,p) - r_p} < 1 + \varepsilon$$



# Approximate Similarity Search

1. relative error approximation (pruning condition)
   - Range and k-NN search queries
2. **good fraction approximation (stop condition)**
   - **K-NN search queries**
3. small chance improvement approximation
4. proximity-based approximation
5. PAC nearest neighbor searching
6. performance trials

# Good fraction approximation

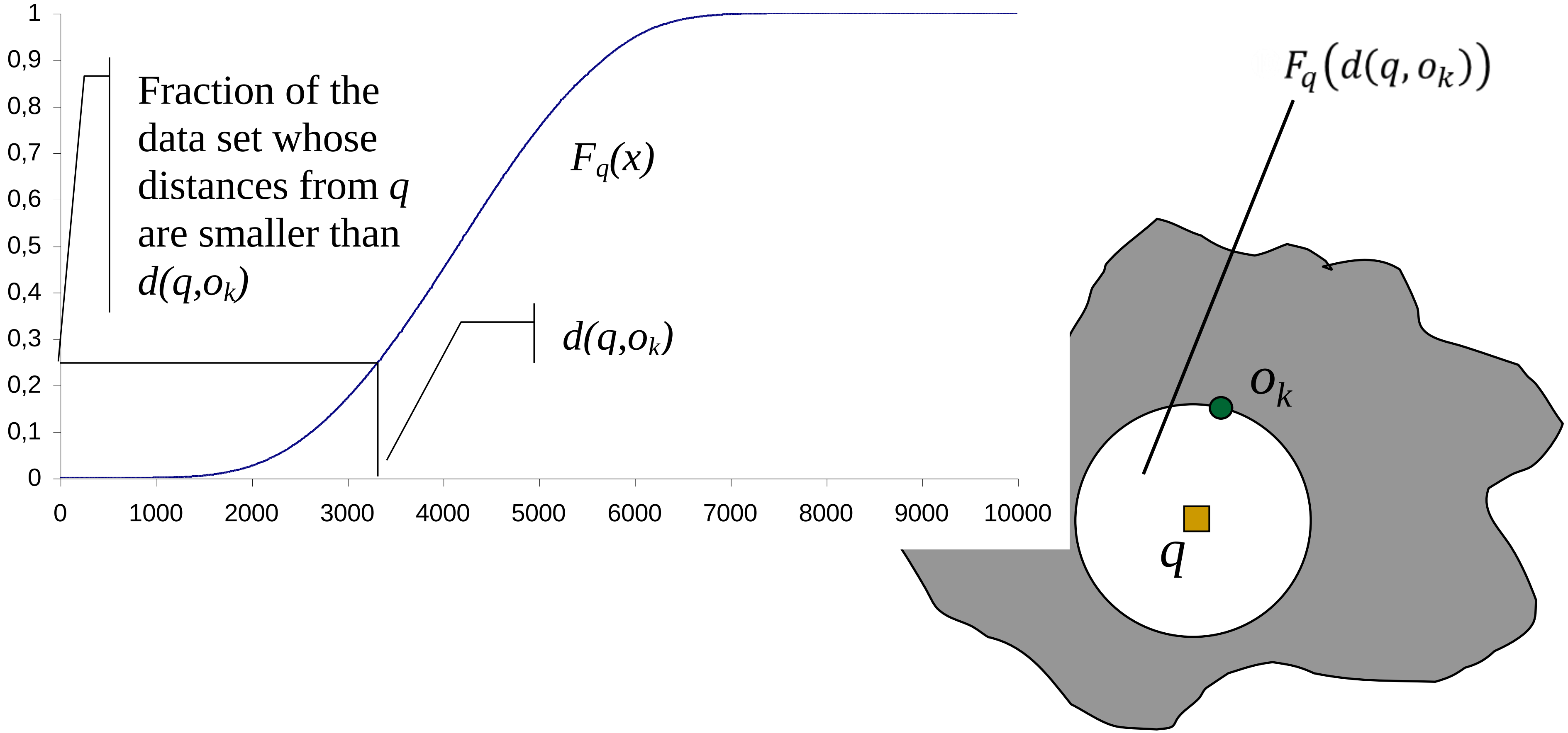
- The *k*-NN algorithm determines the final result by reducing distances of current result set

- When the current result set belongs to a specific fraction of the objects closest to the query, the approximate algorithm stops
  - Example: Stop when current result set belongs to the 10% of the objects closest to the query

# Good fraction approximation

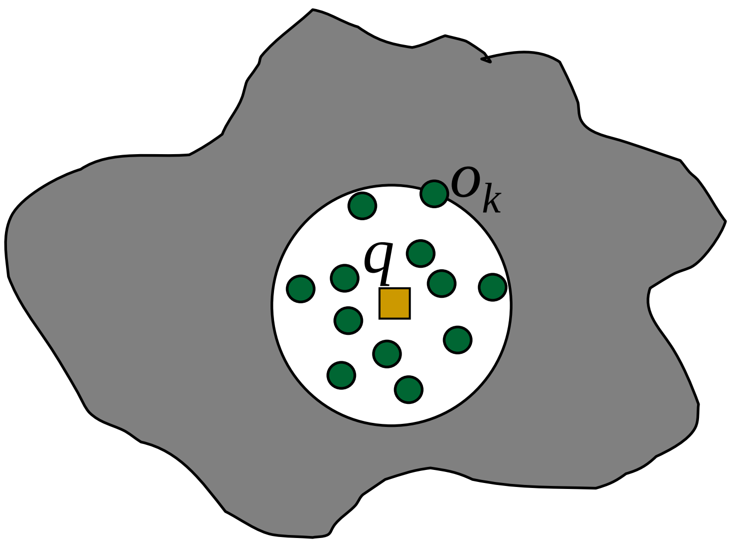
- For this strategy we use the distance distribution defined as
$$F_q(x) = \Pr\left(d(\boldsymbol{o}, q) \le x\right)$$
- The distance distribution $F_q(x)$ specifies what is the probability that the distance of a random object ***o*** from $q$ is smaller than $x$
- It is easy to see that $F_q(x)$ gives, in probabilistic terms, the fraction of the database corresponding to the set of objects whose distance from $q$ is smaller than $x$

# Good fraction approximation



# Good fraction approximation

- When $F_q(d(o_k,q)) < \rho$ all objects of the current result set belong to the fraction $\rho$ of the dataset



# Good fraction approximation

- $F_q(x)$ is difficult to be handled since we need to compute it for all possible queries
- It was proven that the overall distance distribution $F(x)$ defined as follows

$$F(x) = \Pr\left(d(\boldsymbol{o_1}, \boldsymbol{o_2}) \leq x\right)$$

  can be used in practice, instead of $F_q(x)$, since they have statistically the same behaviour.
- $F(x)$ can be easily estimated as a discrete function and it can be easily maintained in main memory

# Approximate Similarity Search

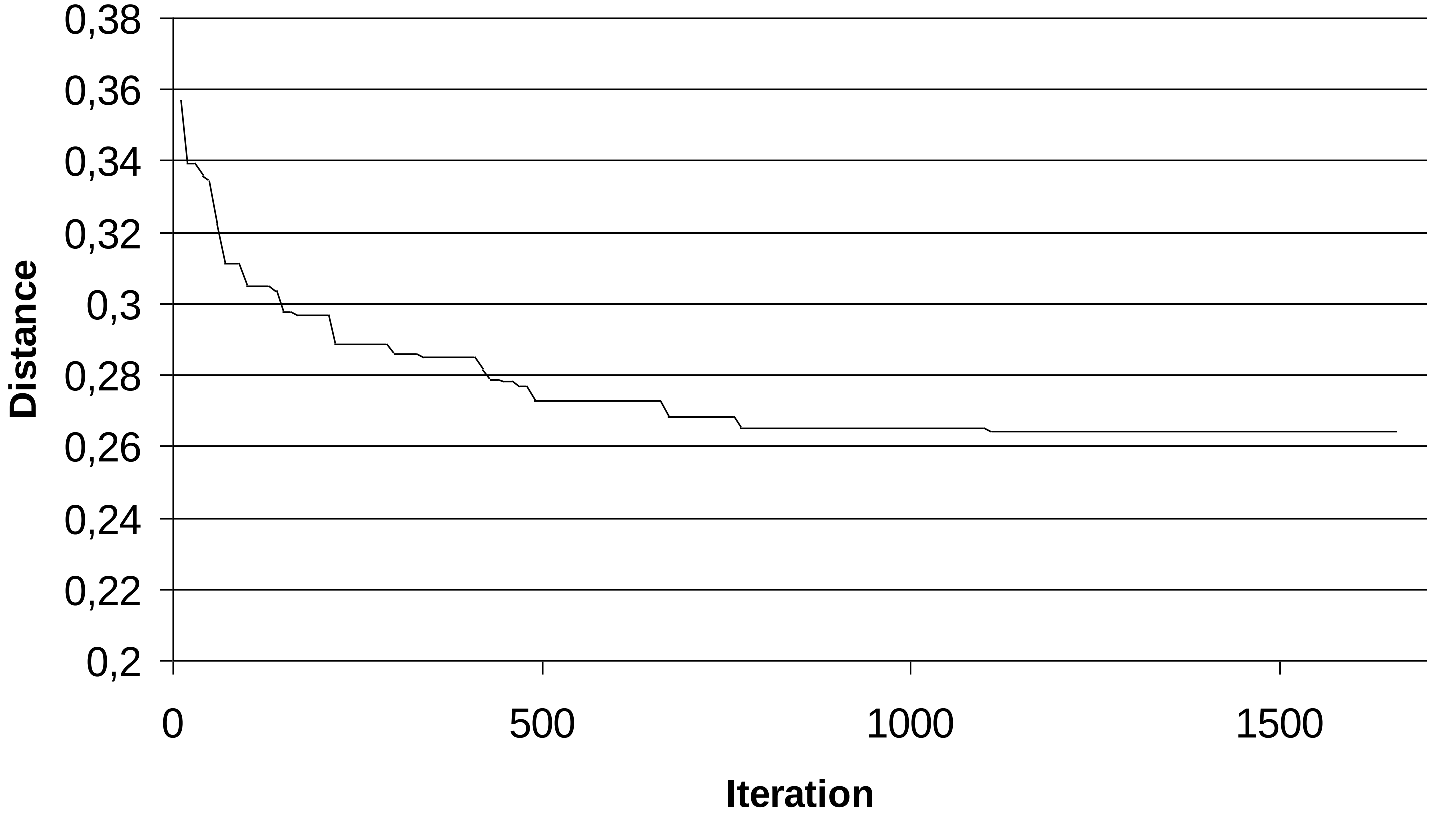
1. relative error approximation (pruning condition)
   - Range and k-NN search queries
2. good fraction approximation (stop condition)
   - K-NN search queries
3. **small chance improvement approximation (stop c.)**
   - **K-NN search queries**
4. proximity-based approximation
5. PAC nearest neighbor searching
6. performance trials

# Small chance improvement approximation

- The M-Tree's *k*-NN algorithm determines the final result by improving the current result set
- Each step of the algorithm the temporary result is improved and the distance of the k-th element decreases
- When the improvement of the temporary result set slows down, the algorithms can stop

# Small chance improvement approximation

$$f(x): \longrightarrow d(q, o_k^A)$$



# Small chance improvement approximation

- Function $f(x)$ is not known a priori.
- A **regression curve** $\varphi(x)$, which approximate $f(x)$, is computed using the **least square method** while the algorithm proceeds
- Through the derivative of $\varphi(x)$ it is possible to decide when the algorithm has to stop

# Small chance improvement approximation

- The regression curve has the following form

$$\varphi(x) = c_1\varphi_1(x) + c_2$$

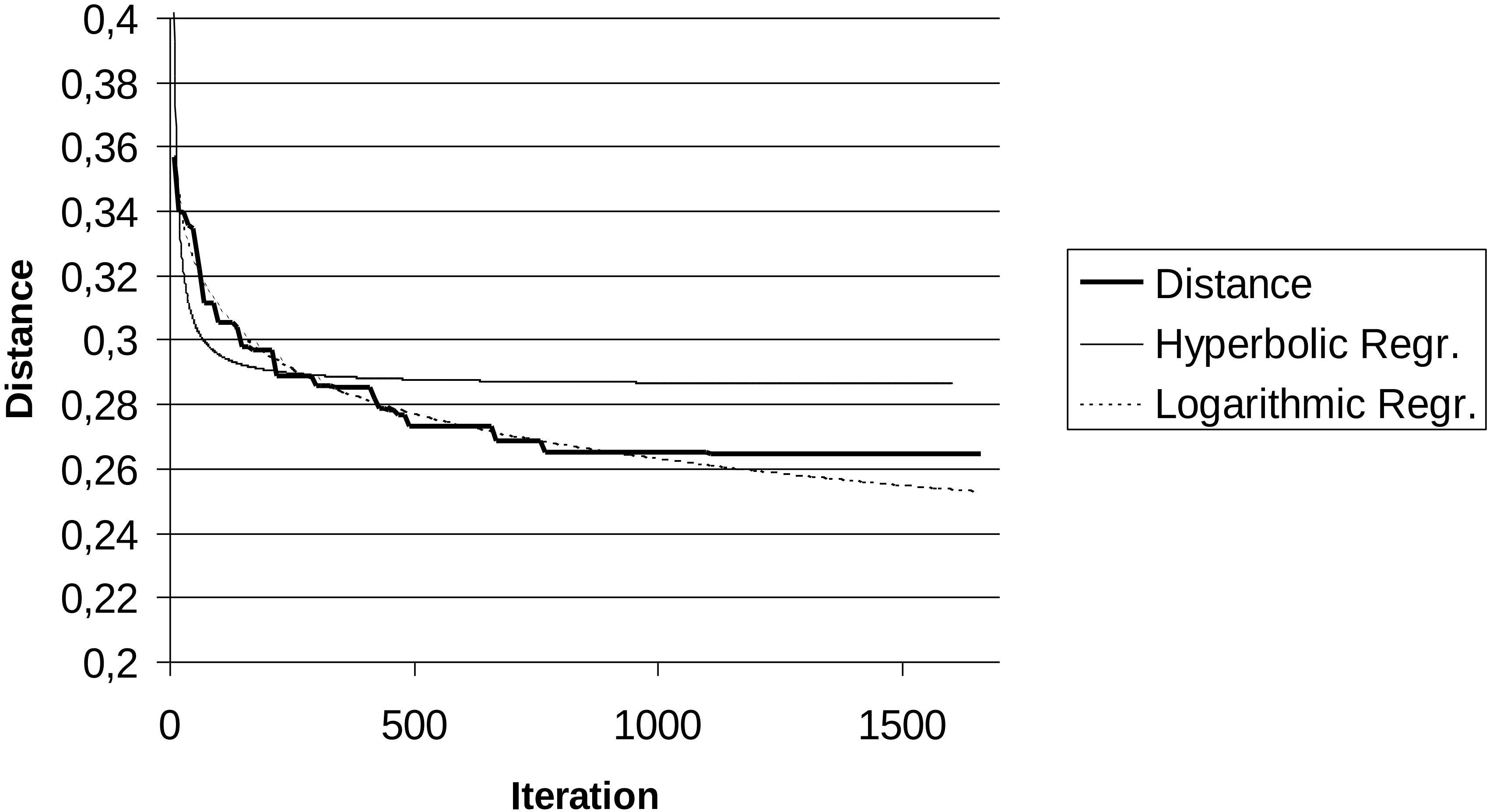
where $c_1$ and $c_2$ are such that

$$\sum_{i=0}^{j}\left(c_1\varphi_1(i) + c_2 - f(i)\right)^2$$

is minimum

- We have used both $\varphi_1(x)=ln(x)$ and $\varphi_1(x)=1/x$

# Regression curves



# Approximate Similarity Search

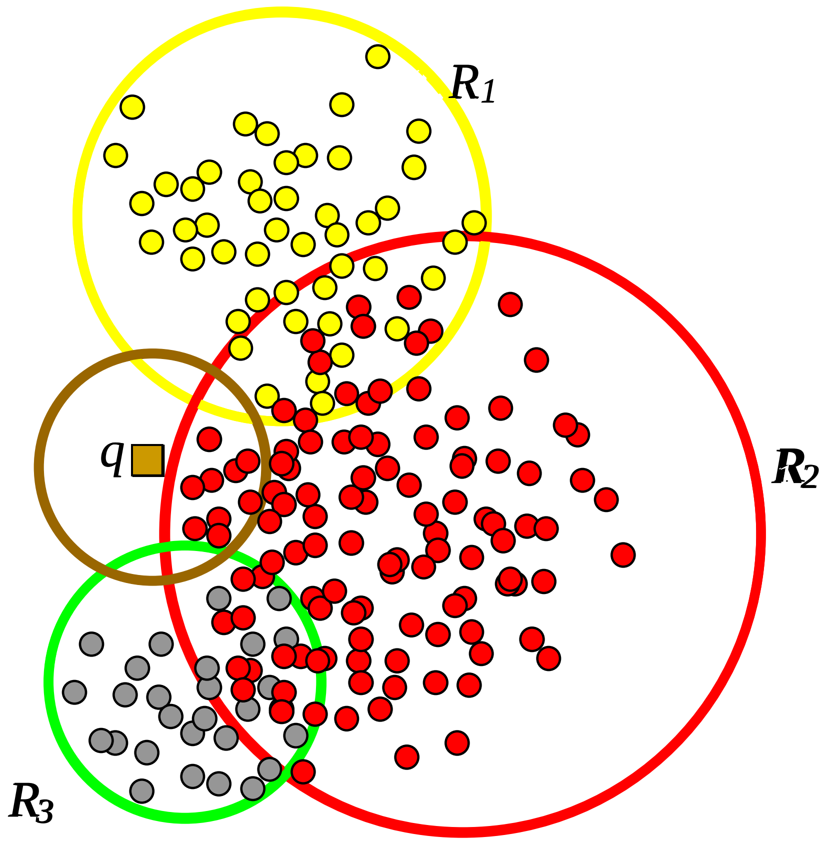
1. relative error approximation (pruning condition)
   - Range and k-NN search queries
2. good fraction approximation (stop condition)
   - K-NN search queries
3. small chance improvement approximation (stop c.)
   - K-NN search queries
4. **proximity-based approximation (pruning cond.)**
   - **Range and k-NN search queries**
5. PAC nearest neighbor searching
6. performance trials

# Proximity-based approximation

- Regions whose probability of containing qualifying objects is below a certain threshold are pruned even if they overlap the query region
  - **Proximity** between regions is defined as the probability that a randomly chosen object appears in both the regions.

- This resulted in an increase of performance of two orders of magnitude both for range queries and nearest neighbour queries

# Proximity-based approximation



# Approximate Similarity Search

1. relative error approximation (pruning condition)
   - Range and k-NN search queries
2. good fraction approximation (stop condition)
   - K-NN search queries
3. small chance improvement approximation (stop c.)
   - K-NN search queries
4. proximity-based approximation (pruning cond.)
   - Range and k-NN search queries
5. **PAC nearest neighbor searching (pruning & stop)**
   - **1-NN search queries**
6. performance trials

# PAC nearest neighbour searching

- It uses the same time a relaxed branching condition and a stop condition
  - The relaxed branching condition is the same used for the relative error approximation to find an $(1+\varepsilon)$-approximate-nearest neighbor
  - In addition it halts prematurely when the probability that we have found the $(1+\varepsilon)$-approximate-nearest neighbor is above the threshold $\delta$
- It can only be used for 1-NN search queries

# PAC nearest neighbour searching

- Let us suppose that then nearest neighbour found so far is $o^A$
- Let $\varepsilon_{act}$ be the actual error on distance of $o^A$

$$\varepsilon_{act} = \frac{d(o^A, q)}{d(o^N, q)} - 1$$

- The algorithm stops if

$$\Pr\{\varepsilon_{act} \geq \varepsilon\} \leq \delta$$

- The above probability is obtained by computing the *distribution of the distance of the nearest neighbor.*

# PAC nearest neighbour searching

- Distribution of the distance of the nearest neighbor in *X* (of cardinality *n*) with respect to *q*:

$$G_q(x) = \Pr\{\exists o \in X : d(q,o) \leq x\} = 1 - (1 - F_q(x))^n$$

- Given that

$$\Pr\{\varepsilon_{act} \geq \varepsilon\} = \Pr\{\exists o \in X : d(q,o^A)/d(q,o) - 1 \geq \varepsilon\} =$$
$$= \Pr\{\exists o \in X : d(q,o) \leq d(q,o^A)/(1+\varepsilon)\} = G_q\left(d(q,o^A)/(1+\varepsilon)\right)$$

- The algorithm halts when

$$G_q\left(d(q,o^A)/(1+\varepsilon)\right) \leq \delta$$

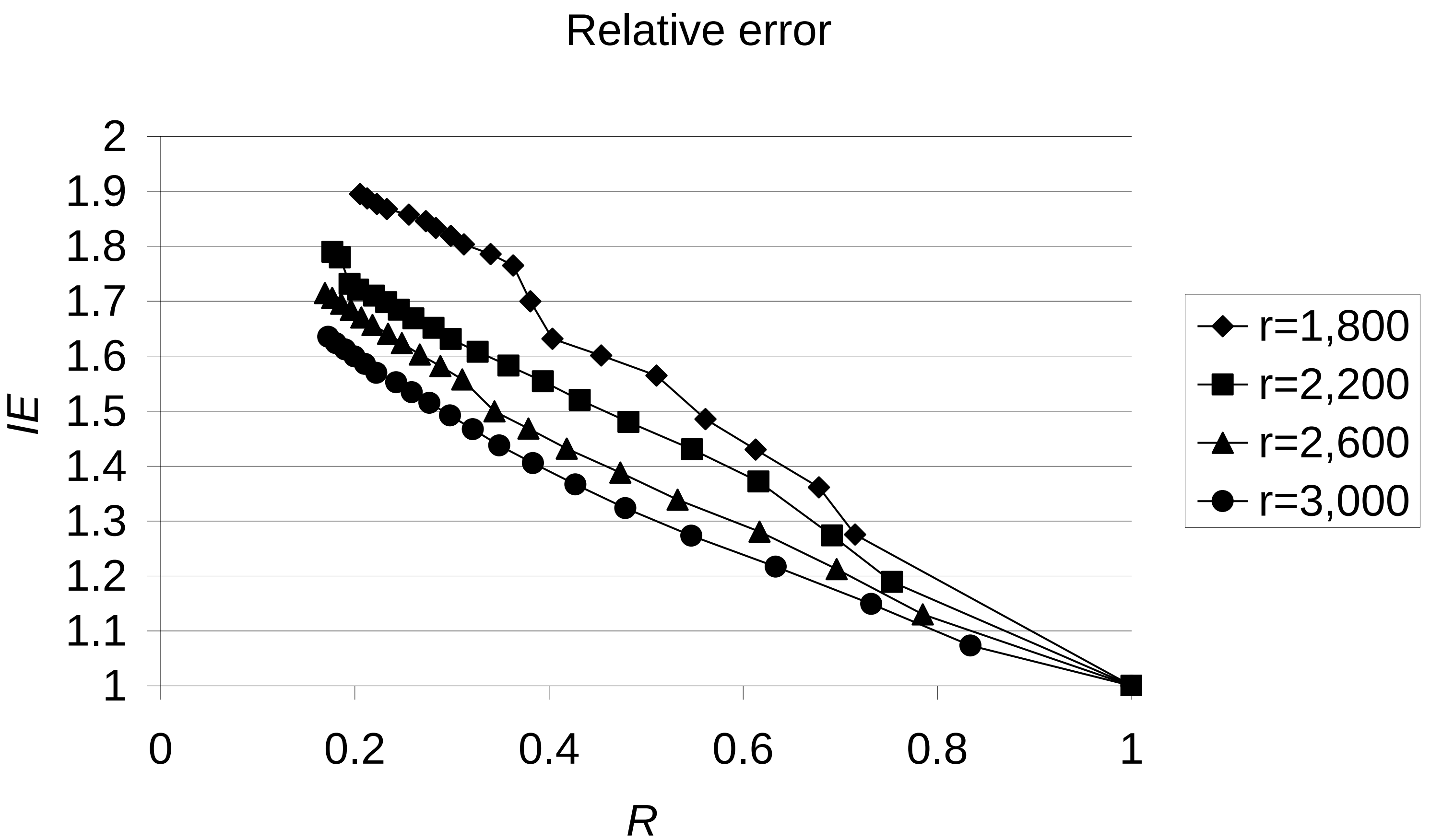
# Approximate Similarity Search

1. relative error approximation (pruning condition)
   - Range and k-NN search queries
2. good fraction approximation (stop condition)
   - K-NN search queries
3. small chance improvement approximation (stop c.)
   - K-NN search queries
4. proximity-based approximation (pruning cond.)
   - Range and k-NN search queries
5. PAC nearest neighbor searching (pruning & stop)
   - 1-NN search queries
6. **performance trials**

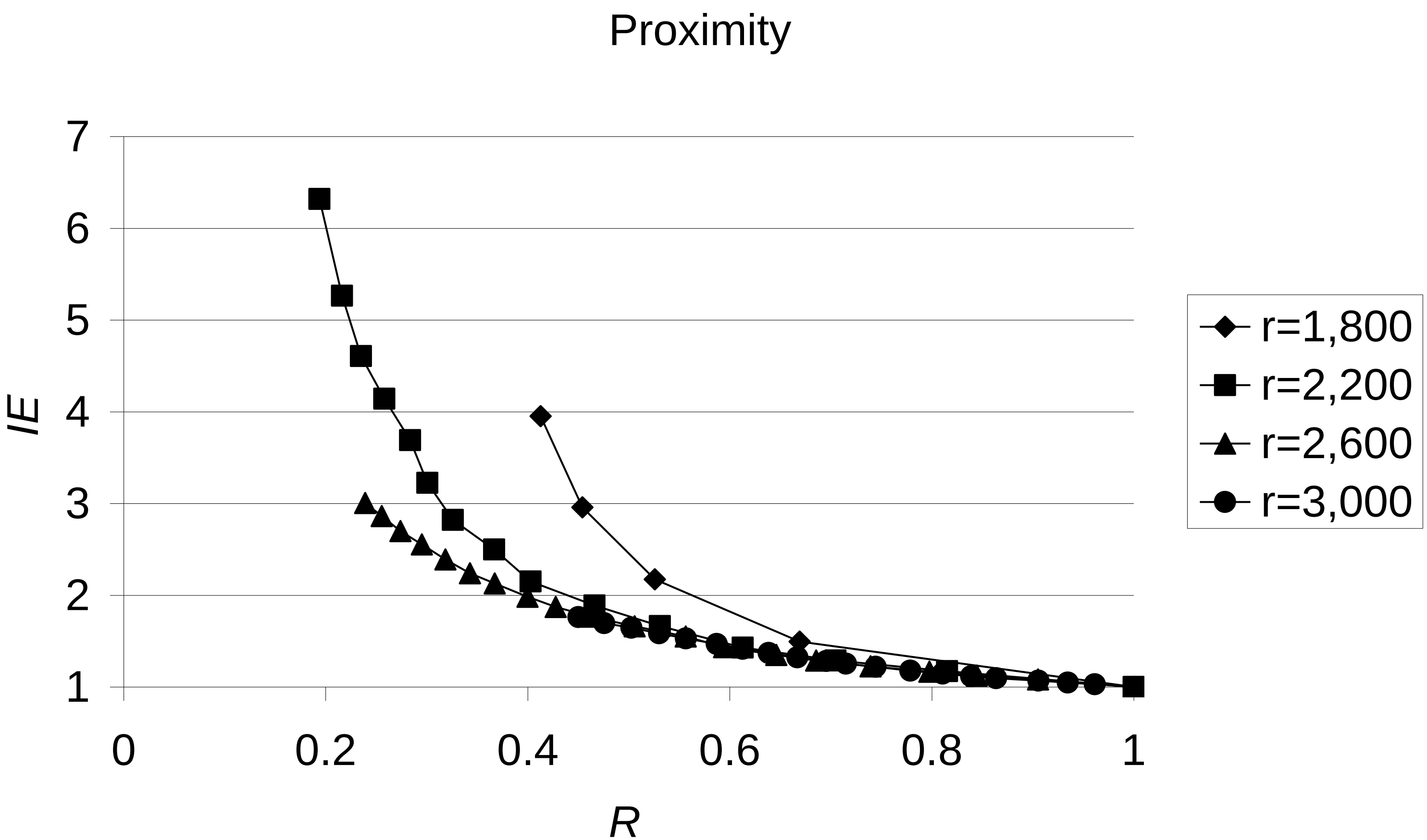
# Comparisons tests

- Tests on a dataset of 11,000 objects
  - Objects are vectors of 45 dimensions
- We compared the five approximation approaches
  - Range queries tested on the methods:
    - Relative error
    - Proximity
  - Nearest-neighbors queries tested on all methods

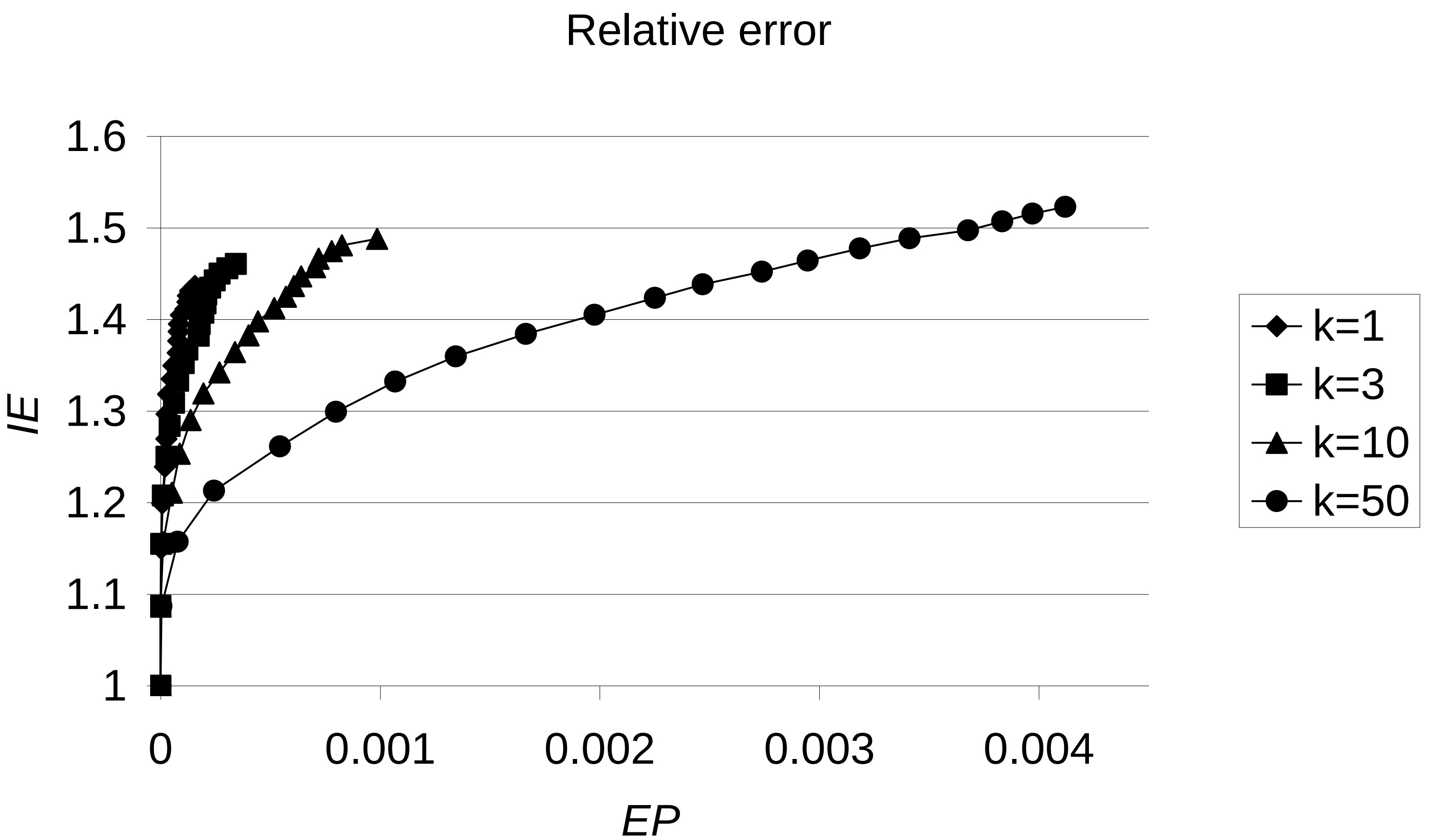
# Comparisons: range queries



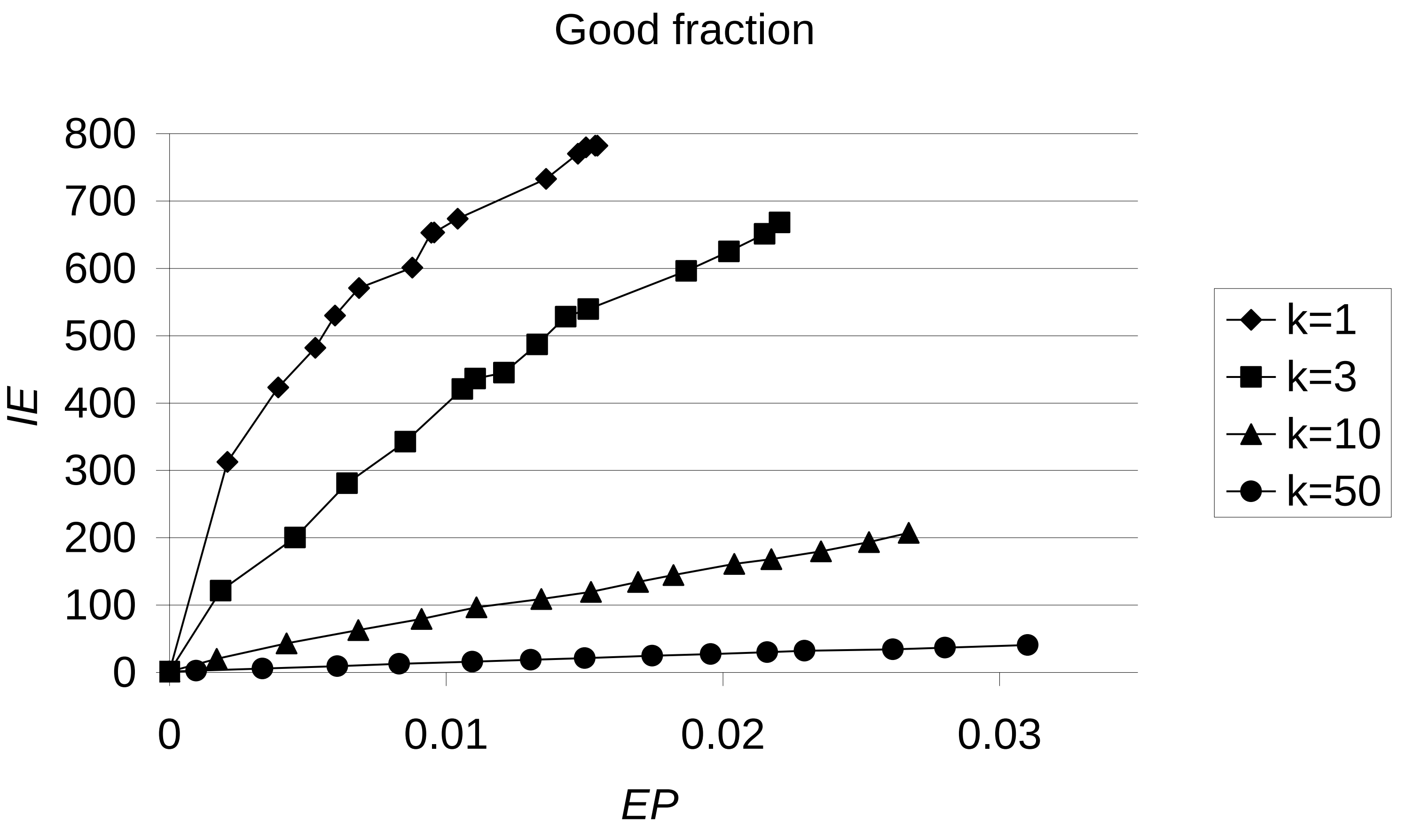
# Comparisons: range queries



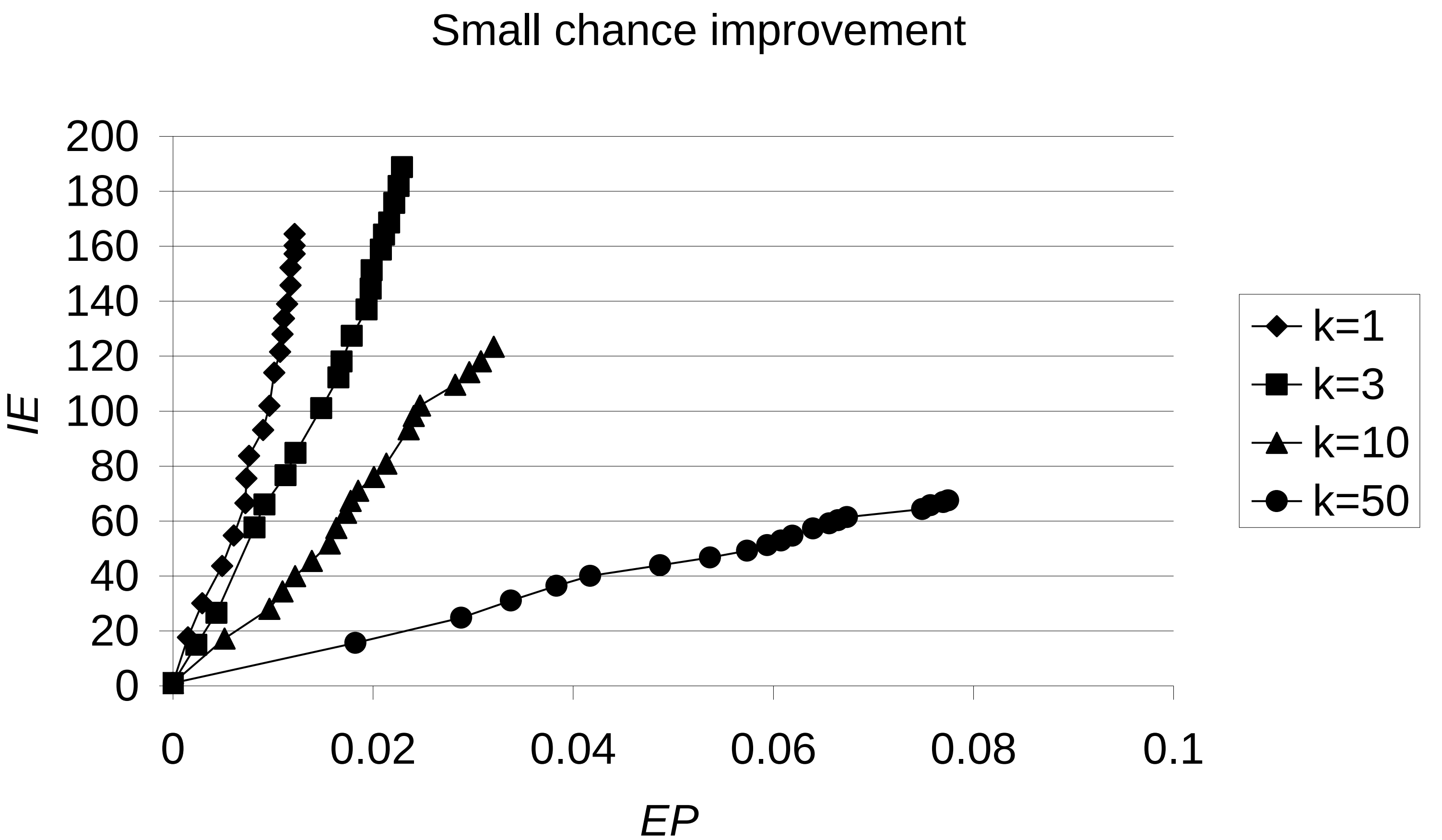
# Comparisons NN queries



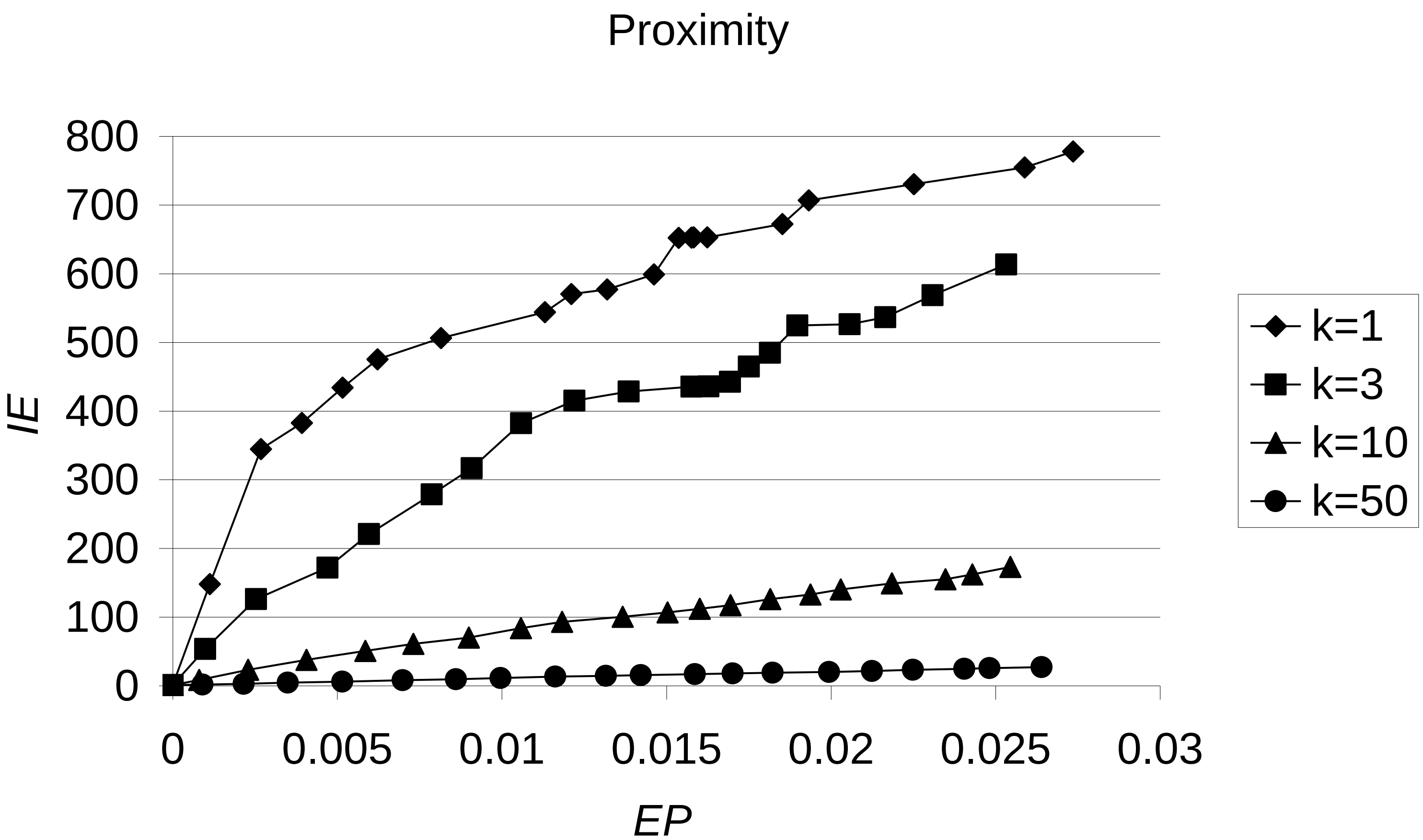
# Comparisons NN queries



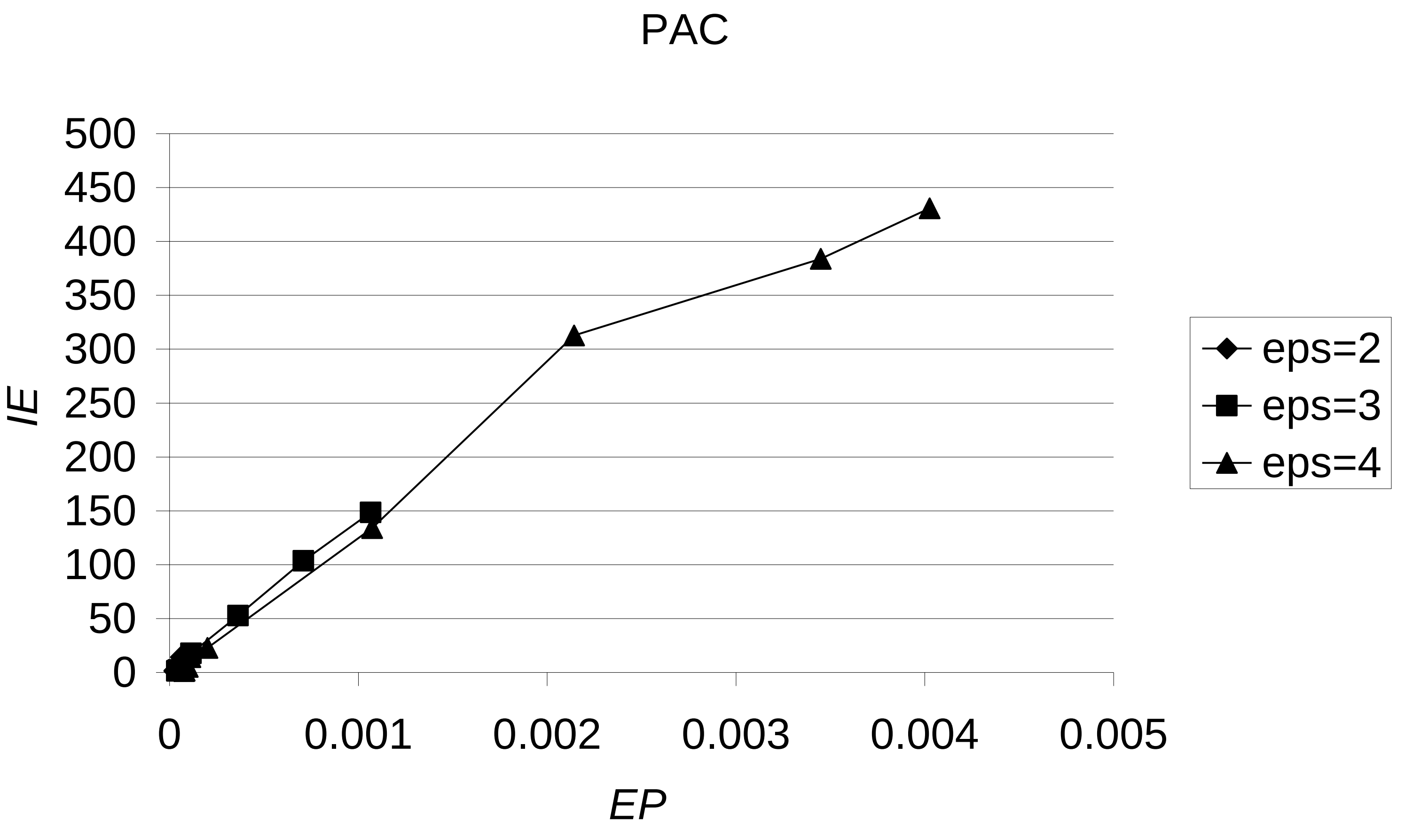
# Comparisons NN queries



# Comparisons NN queries



# Comparisons NN queries



# Conclusions: Approximate similarity search in metric spaces

- These techniques for approximate similarity search can be applied to generic metric spaces
  - Vector spaces are a special case  of metric space.
- High accuracy of approximate results are generally obtained with high improvement of efficiency
  - Best performance obtained with the good fraction approximation methods
  - The proximity based is a bit worse than good fraction approximation but can be used for range queries and *k*-NN queries.