Indexing hierarchical data using *pq* grams

Nitish Srivastava Varunesh Mishra

Department of Computer Science and Engineering, IIT Kanpur

November 7, 2009

- The task:build an optimal indexing technique for tree data.
- The data to be indexed is not relational.
- No absolute ordering property.

Tree edit distance

- Counts the minimum number of insert, delete and relabel operations required to convert one tree into another
- Takes almost $O(m^2 n^2)$ time.
- Too inefficient for large trees.

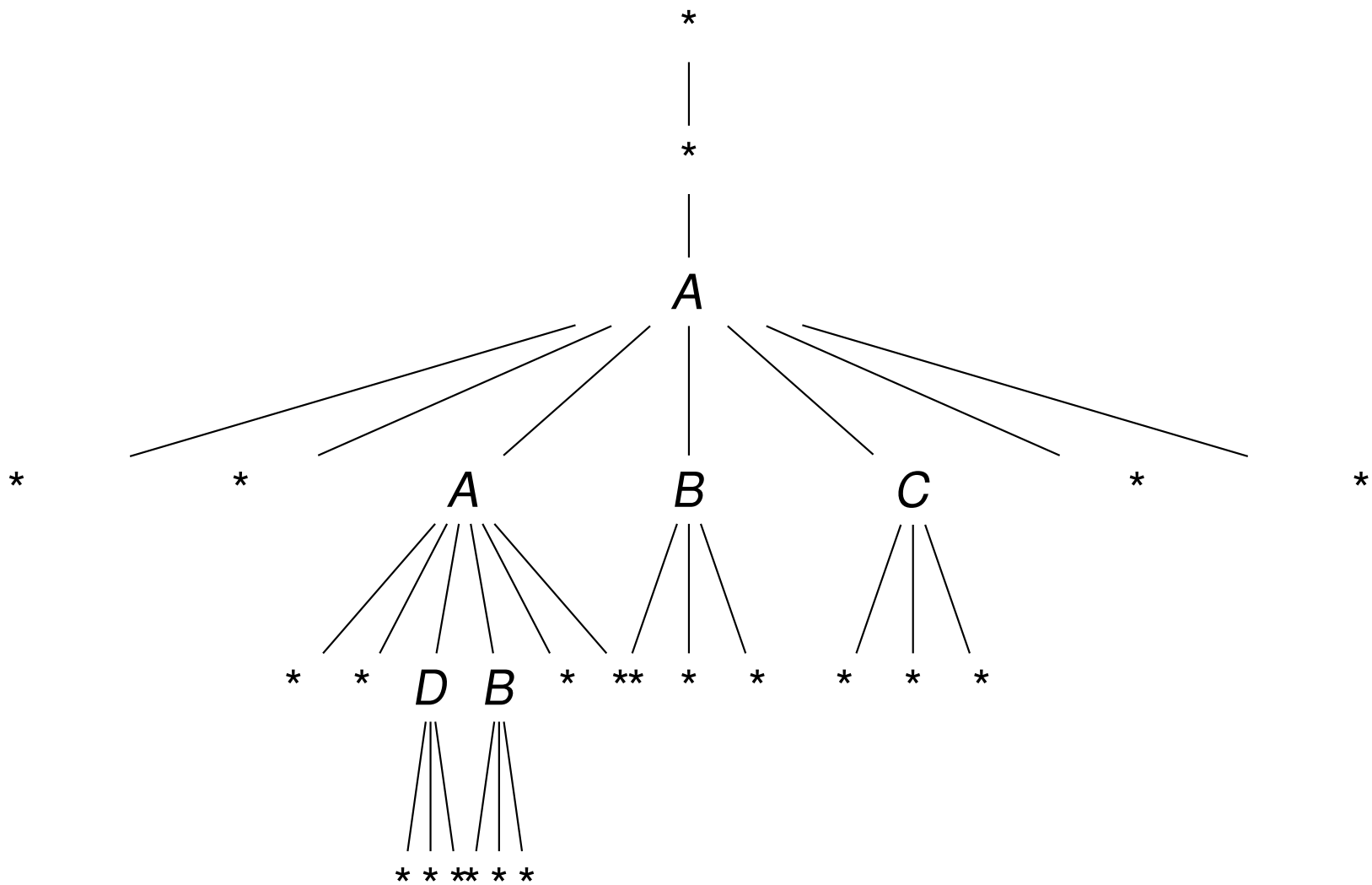
pq gram distance

- A distance measure that compares trees on the basis of number of common subtrees.
- Can be tuned to set the trade-off between importance given to structure vs. data.
- Fast computation time.
- Requires high storage space.

We try to build an efficient index structure and also determine the exact nature of the structure vs data tradeoff.

pq gram distance

Extended Tree for $p = 3$, $q = 3$



pq gram profile

Multiset of pq grams

$$\{*, *, A, *, *, A\} \quad \{*, *, A, *, A, B\}$$
$$\{*, *, A, A, B, C\} \quad \{*, *, A, B, C, *\}$$
$$\{*, *, A, C, *, *\} \quad \{*, A, A, *, *, E\}$$
$$\{*, A, A, *, E, B\} \quad \{*, A, A, E, B, *\}$$
$$\{*, A, A, B, *, *\} \quad \{*, A, B, *, *, *\}$$
$$\{*, A, C, *, *, *\} \quad \{A, A, E, *, *, *\}$$
$$\{A, A, B, *, *, *\}$$

pq gram distance

Definition (pq gram distance (Augsten [1]))

pq gram distance between two trees T_1 and T_2 is defined as

$$d^{pq}(T_1, T_2) = |\mathcal{I}_1 \uplus \mathcal{I}_2| - 2.|\mathcal{I}_1 \boxplus \mathcal{I}_2|$$

where $\mathcal{I}_i$ is the pq gram profile for tree T_i.

Choosing optimal p, q values

The performance of pq distance depends on the values of p and q.

- Increasing p and q values makes the profile more *rigid*.
- Decreasing them makes the profile insensitive to structure.
- Increasing p relative to q increases importance given to ancestors over children.

Probabilistic analysis

To study the dependence of pq gram distance on p and q

- we define a procedure for generating a random tree T.
- analytically find the expected value and variance in pq gram distance between two random trees.

Then we try to validate the results with experimental observations. Another way to analyze the dependence is by generating one random tree T_1 and then performing insert, delete and relabel operations on it to generate T_2. The variation in $d^{pq}(T_1, T_2)$ with change in p and q is observed.

Probabilistic Analysis

Generating a random tree Given a label set Σ, a random tree T^k is constructed as follows

1. $T^k = \{\text{root}\}, i \leftarrow 0$
2. Choose a leaf node v from T^k uniformly randomly.
3. Create ξ nodes by choosing ξ from $\mathcal{N}(\mu, \sigma)$. Add them to T^k as children of v.
4. $i++$
5. if $i < k$ goto 2
6. Assign a label to each node of T^k by sampling uniformly randomly from the set of labels Σ.

Probabilistic Analysis

Let $l_{h,t}$ and $n_{h,t}$ be random variables.
$l_{h,t}$ - number of leaves in T^k at height h.
$n_{h,t}$ - number of internal nodes in T^k at height h.
Let v be drawn from $\mathcal{N}(\mu, \sigma)$. Then,

$$l_{h,t+1} = l_{h,t} - \frac{l_{h,t}}{\sum_{h=0}^{\infty} l_{h,t}} + v\frac{l_{h-1,t}}{\sum_{h=0}^{\infty} l_{h,t}}$$

$$n_{h,t+1} = n_{h,t} + \frac{l_{h,t}}{\sum_{h=0}^{\infty} l_{h,t}}$$

Probabilistic Analysis

After t iterations

$$\sum_{h=0}^{\infty} n_{h,t} = t$$

$$\sum_{h=0}^{\infty} (l_{h,t} + n_{h,t}) = 1 + tv \Rightarrow \sum_{h=0}^{\infty} l_{h,t} = (v-1)t + 1$$

We finally get the following recursive equations

$$n_{h,t+1} = n_{h,t} + \frac{l_{h,t}}{(v-1)t+1}$$

$$l_{h,t+1} = l_{h,t} + \frac{v l_{h-1,t} - l_{h,t}}{(v-1)t+1}$$

$$l_{0,0} = 1 \quad , \quad n_{0,0} = 0$$

Probabilistic Analysis

$$l_{h,t+1} - l_{h,t} = \frac{v l_{h-1,t} - l_{h,t}}{(v-1)t+1}$$

$$\Rightarrow ((v-1)t+1)(l_{h,t+1} - l_{h,t}) = v l_{h-1,t} - l_{h,t}$$

Let $X(z_1, z_2) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} l[h,t] z_1^{-m} z_2^{-n}$ be the Z-tranform of $l[h,t]$.

Then

$$(v-1)z_2(z_2 \frac{\partial X}{\partial z_2} + X) - (v-1)z_2 \frac{\partial X}{\partial z_2} + z_2 X = \frac{k}{z_1} X$$

This gives

$$\frac{\partial X}{\partial z_2} = \frac{vX}{(k-1)z_2(z_2-1)} \left(\frac{1}{z_1} - z_2 \right)$$

Probabilistic Analysis

$$X = \left(\left(1 - \frac{1}{z_2}\right)^{1/z_1} \frac{1}{z_2 - 1}\right)^{\frac{v}{v-1}}$$

Using this we can compute $l_{h,t}$ for any h ,t. We also had

$$n_{h,t+1} = n_{h,t} + \frac{l_{h,t}}{(v-1)t + 1}$$

Therefore,

$$n_{h,T} = \sum_{t=0}^{T} \frac{l_{h,t}}{(v-1)t + 1}$$

Probabilistic analysis

The most general form of a label-tuple of a pq gram is

$$\lambda = (*^a v_1, v_2, \ldots, v_b, *^c, w_1, w_2, \ldots, w_d, *^e)$$

where $a + b = p$, $c + d + e = q$, $a, b, c, d, e \geq 0$

$$\begin{aligned} \lambda^p &= (*^a v_1, v_2, \ldots, v_b) \\ \lambda^q &= (*^c, w_1, w_2, \ldots, w_d, *^e) \end{aligned}$$

$$P(\lambda) = P(\lambda^p)P(\lambda^q)$$

Probabilistic analysis

δ_k be a random variable equal to the pq gram distance between two random trees.

- Find how many times each λ occurs in T^k. Let this be η_i.
- To find this we find how many times λ occurs in T^k with anchor node *at height h*.

$$\eta_i = \sum_{h=0}^{k} \eta_{i,h}$$

- To find $\eta_{i,h}$ we find $\eta_{i,h,j}$ the number of times λ_i occurs *at a given node position j* at height h.

$$\eta_{i,h} = \sum_{j=0}^{\infty} \eta_{i,h,j}$$

Probabilistic analysis

Probability of occurence of λ^p(p part of λ
If $a > 0$ then λ^p can occur only at height $h = b - 1$. Therefore,

$$P(\lambda^p | a > 0) = \begin{cases} 0 & h \neq b - 1 \\ \dfrac{1}{|\Sigma|^b} & h = b - 1 \end{cases}$$

Else it can occur anywhere.

$$P(\lambda^p | a = 0) = \frac{1}{|\Sigma|^p}$$

Combining the above,

$$P(\lambda^p) = \frac{1}{|\Sigma|^b}(\delta_{a,0} + \overline{\delta_{a,0}}\delta_{h,b-1})$$

Probabilistic analysis

Probability of occurence of λ^q(q part of λ) If the anchor node is a leaf node then d must be 0, ie

$$P(\lambda^q | v_b \text{is a leaf node}) = \begin{cases} 0 & d \neq 0 \\ 1 & d = 0 \end{cases}$$

Let ξ be a random variable sampled from $\mathcal{N}(\mu, \sigma)$, rounded to the nearest integer x If the anchor node is not a leaf node, d must be less than the number of children of the node (ξ).

$$P(\lambda^q | v_b \text{is not a leaf node}) = \begin{cases} 0 & d > \xi \\ \dfrac{1}{|\Sigma|^d} & l \leq \xi \end{cases}$$

Probabilistic analysis

The above can be combined as,

$$\begin{aligned}
P(\lambda^q) &= \frac{1}{|\Sigma|^d}\left(\delta_{d,0}P(leaf|h) + (1 - P(leaf|h))\,P\left(d \leq \xi\right)\right) \\
E[\eta_{i,h,j}] &= P(\lambda^p)P(\lambda^q) \\
&= \frac{1}{|\Sigma|^{b+d}}(\delta_{a,0} + \overline{\delta_{a,0}}\delta_{h,b-1})(\delta_{d,0}P(leaf|h) \\
&\quad +(1 - P(leaf|h))P(d \leq \xi))) \\
Var[\eta_{i,h,j}] &= P(\lambda)\,(1 - P(\lambda))^2 + (1 - P(\lambda))\,(0 - P(\lambda))^2 \\
&= P(\lambda)\,(1 - P(\lambda)) \\
&= E[\eta_{i,h,j}](1 - E[\eta_{i,h,j}])
\end{aligned}$$

We can use the results of the tree analysis to find $P(leaf|h)$. Then using the statistics for $l_{h,t}$ and $n_{h,t}$ computed earlier, we build up from

$$\eta_{i,h,j} \rightarrow \eta_{i,h} \rightarrow \eta_i$$

Probabilistic analysis

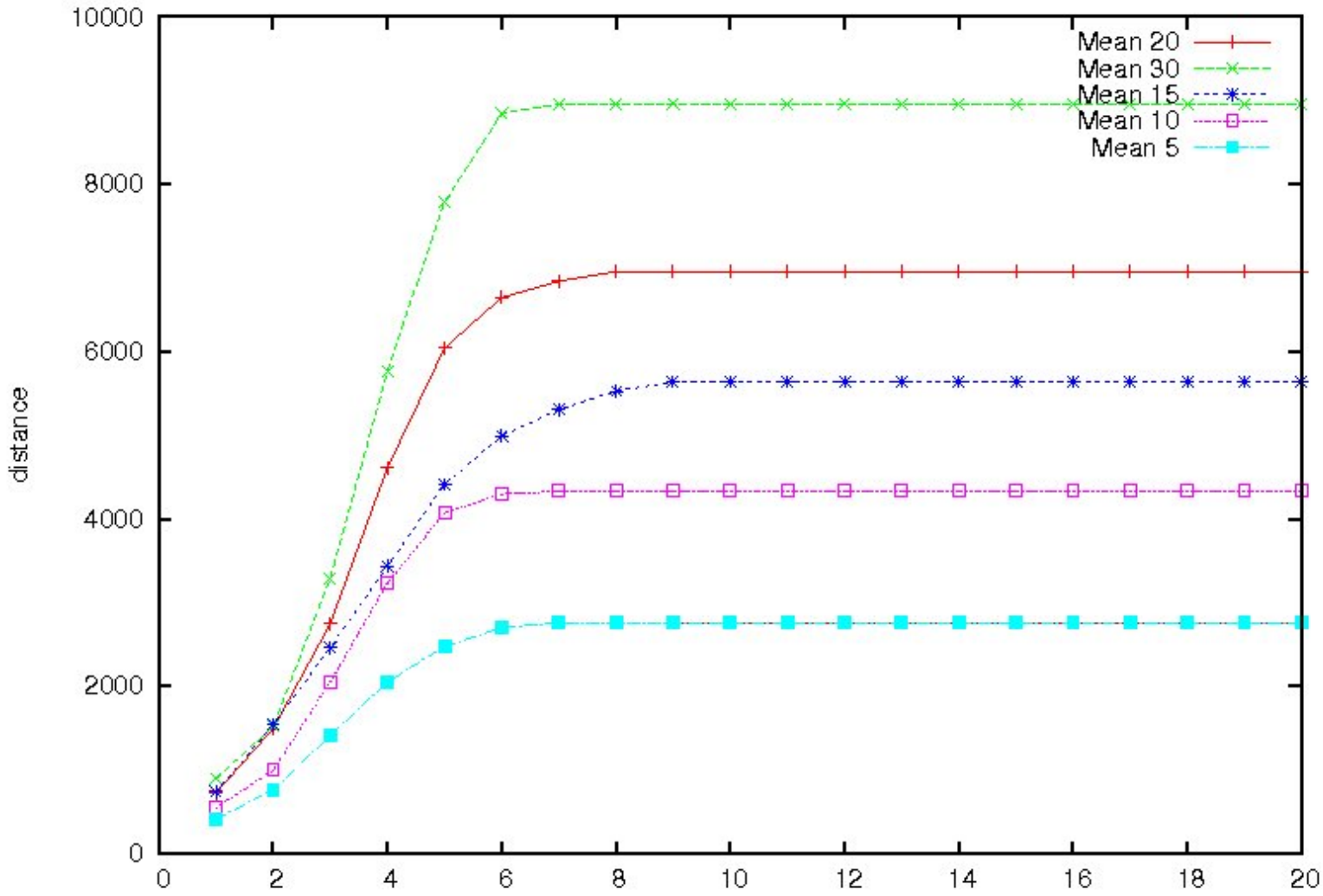
$$d^{pq}(T_1, T_2) = |\mathcal{I}_1 \uplus \mathcal{I}_2| - 2.|\mathcal{I}_1 \boxplus \mathcal{I}_2|$$

Therefore,

$$E[d^{pq}(T_1, T_2)] = 2\sum_i E[\eta_i] - 2\sum_i E[min(\eta_{i_1}, \eta_{i_2})]$$

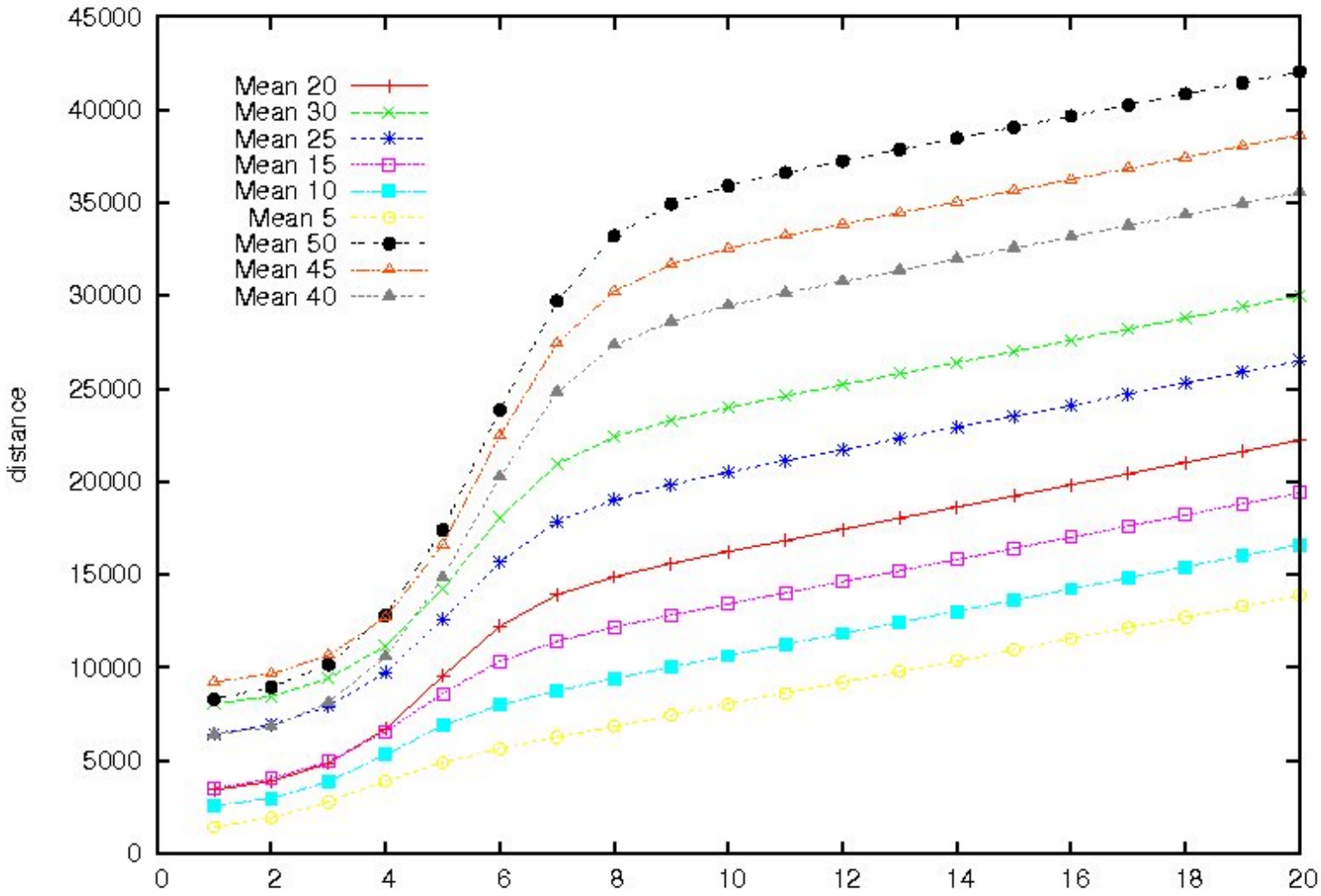
Results obtained

Distance between two random trees varying p for different values of μ



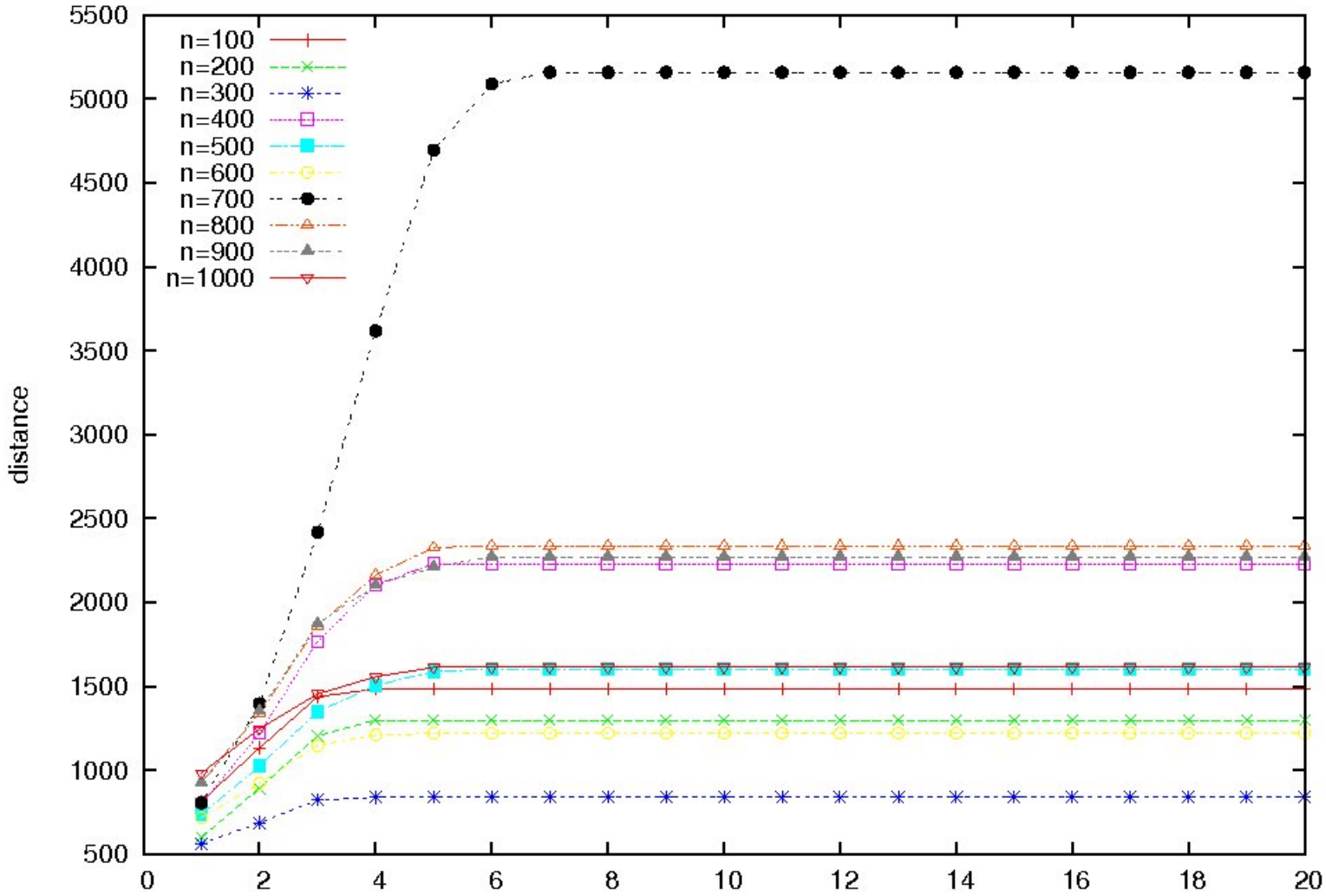
Results obtained

Distance between two random trees varying q for different values of μ



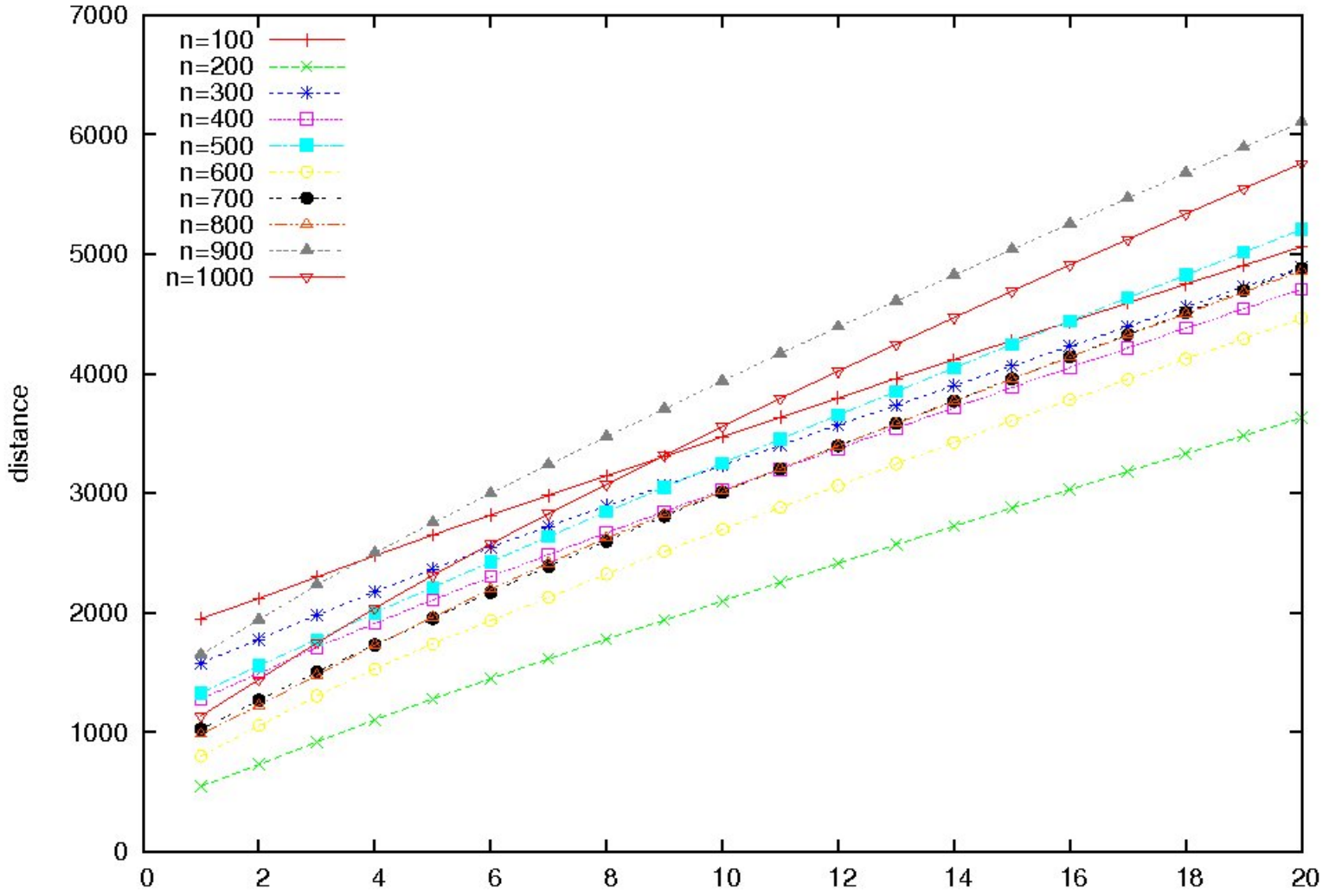
Results obtained

Distance between one random tree and another generated by random edit operations, varying p for different values of μ



Results obtained

Distance between one random tree and another generated by random edit operations, varying q for different values of μ



Index structure for *pq* grams

- In order to use *pq* gram distance, *pq* grams for each tree must be stored in an efficient index strcuture.
- We propose a reference-based indexing scheme based on [2].
- Select certain *good* trees in the database as references
- Reference trees are selected using a maximum variance heuristic
- Each tree in the database is assigned a subset of these reference trees.

Reference selection using maximum variance

/*Input:Sequence database S, with $|S| = N$.
Number of references m.
Cutoff percentage $perc$.
Length of a sequence L.
Output:Set of references $V = \{v_1, v_2, \ldots, v_m\}$*/

1. $V = \{\}$. /* Initialize */
2. **For** each $s_i \in S$ **do**
   (a) Select sample set of sequences, $S' \subset S$.
   (b) Compute $D_i = \{ED(s_i, s_j) \mid \forall s_j \in S'\}$.
   (c) Compute mean μ_i and variance σ_i of the distances in D_i.
3. $w = L.perc$.
4. Sort the N sequences in descending order of their variances.
5. **While** $|V| < m$ **do**
   (a) $V = V \cup s_1$.
   (b) $S = S - \{s_j\}$, $\forall s_j \in S$ with $ED(s_1, s_j) < (\mu_1 - w)$ or $ED(s_1, s_j) > (\mu_1 + w)$. /* Remove sequences close to or far away from the new reference */
6. Return of set of reference sequences, V.

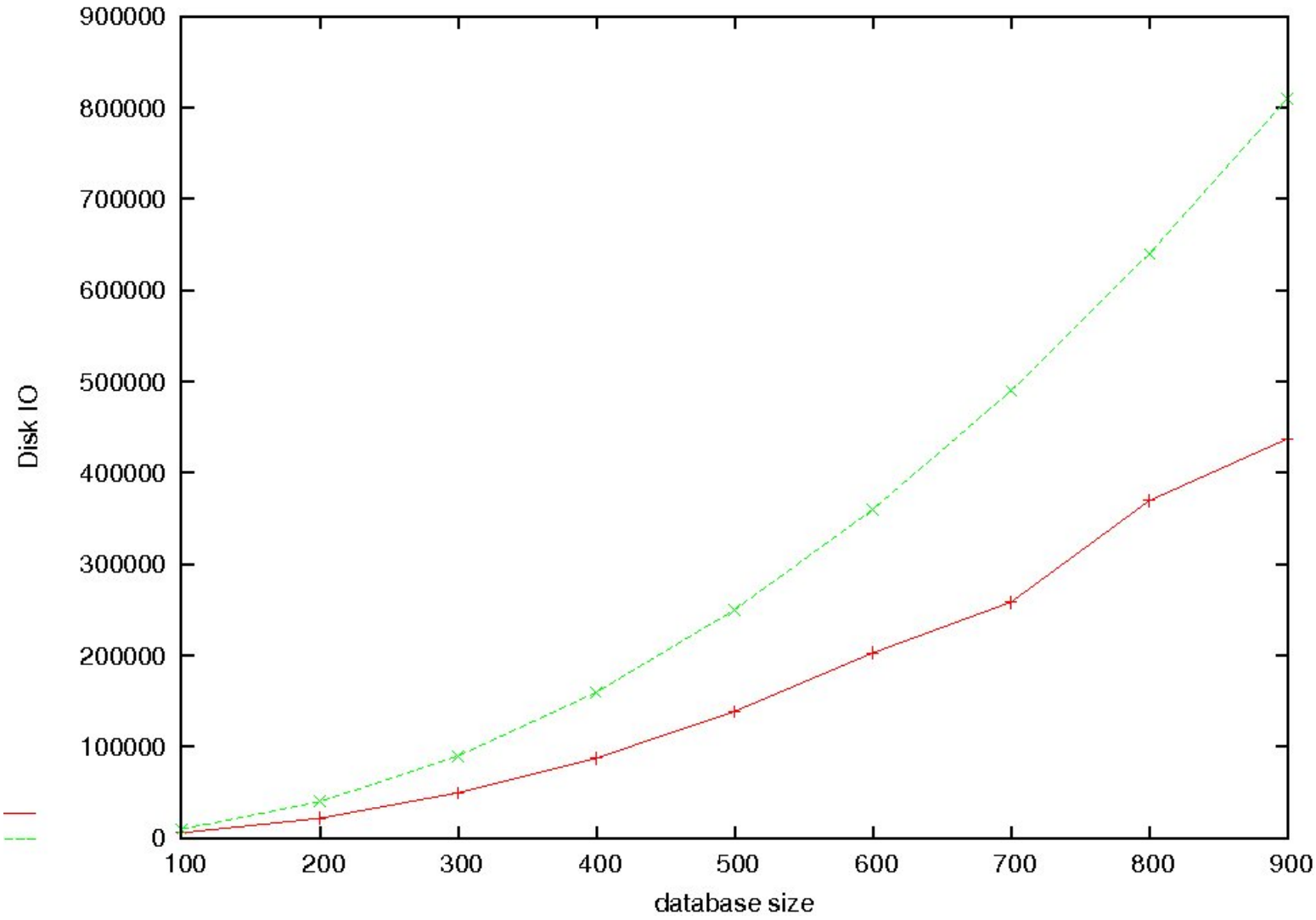
Assignment of references

/*Input:Sequence database S,$|S|$ = N.
Reference set V,$|V|$ = m.
Sequence queries Q,$|Q|$ = q.
References per sequence k.
Output:$E = \{E_1, E_2, \ldots, E_N\}$,
E_s is assigned to sequence $s \in S$.
*/

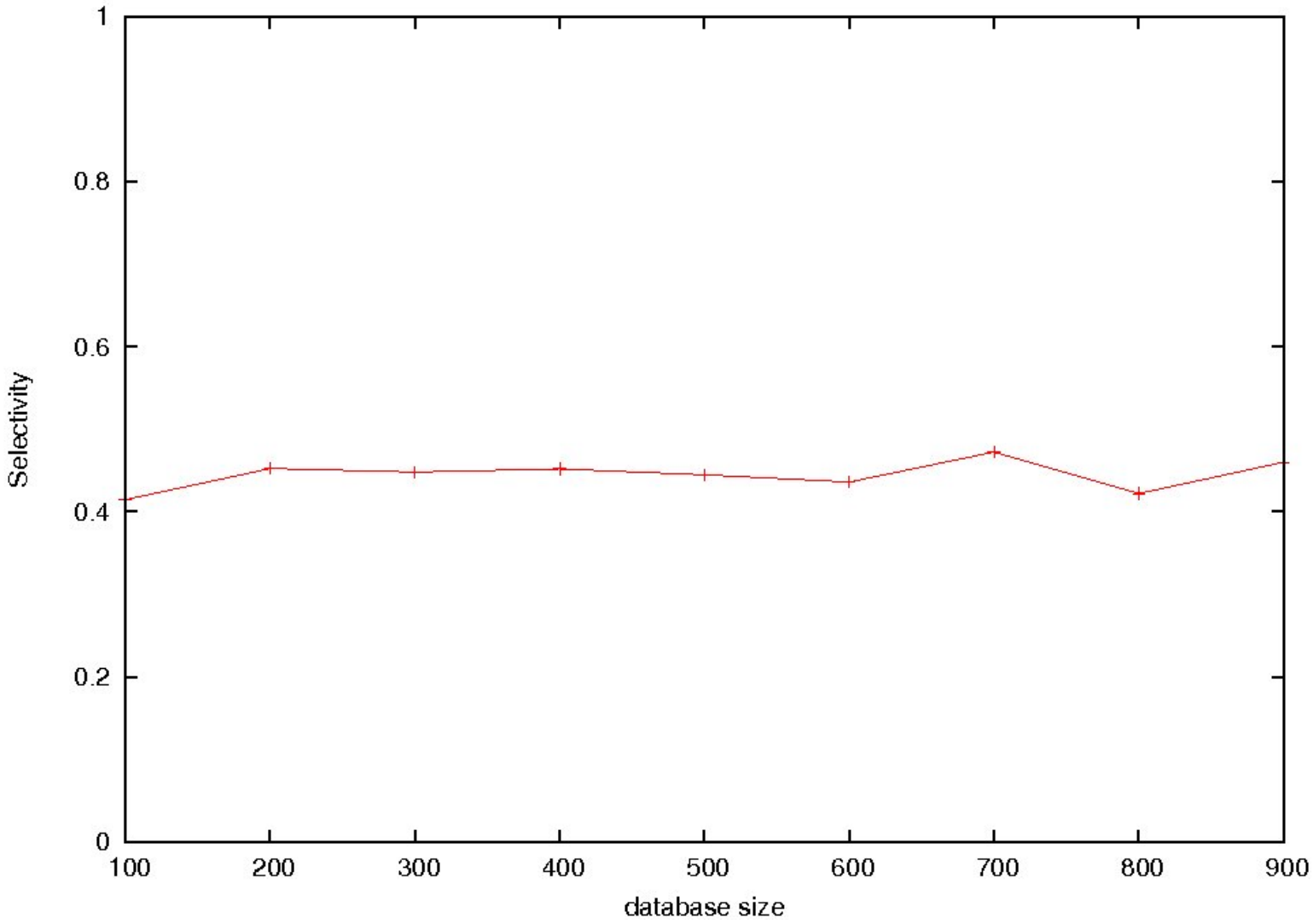
1. $G[i] = 0, 1 \leq i \leq m$. /* Total gain from each reference $v_i \in V$ */
2. $E_i = \{\}, 1 \leq i \leq N$. /*Initialize reference set of each sequence*/
3. **For** each $s \in S$ **do**
   - **Repeat**
     (a) $Vcount[i] = 0, 1 \leq i \leq m$. /* Initialize gain for $[v_i, s]$ pair */
     (b) **For** all $[v, Q_j]$, $\forall v \in V$ and $\forall Q_j \in Q$ **do**
     – **If**(PRUNE(s, Q_j, v)) **do** $Vcount[v]$++.
     (c) Let $e = argmax_x(Vcount[x])$.
     (d) $G[e]$+= $Vcount[e]$.
     (e) $V = V - \{e\}$.
     (f) $E_s = E_s \cup \{e\}$.
     (g) Remove from Q queries for which s is pruned with reference e.

     Until $|E_s| = k$.
   - Re-insert all deleted entries from sets V and Q.
4. **For** all $v \in V$ **do**
   - **If**$(G[v] \leq |Q|)$ $V = V - \{v\}$.
5. Update the reference sets E_s $\forall s \in S$.

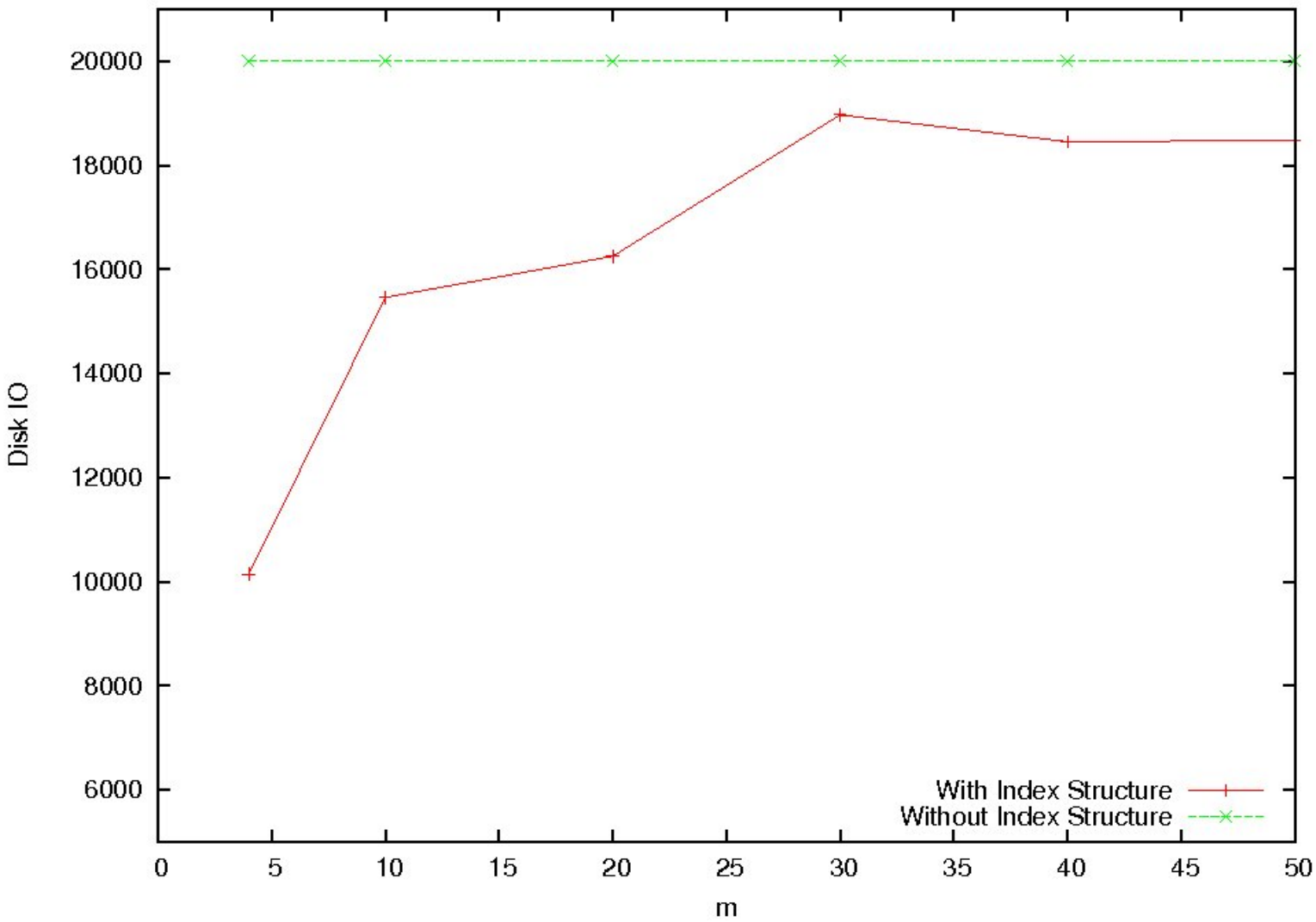
Results obtained



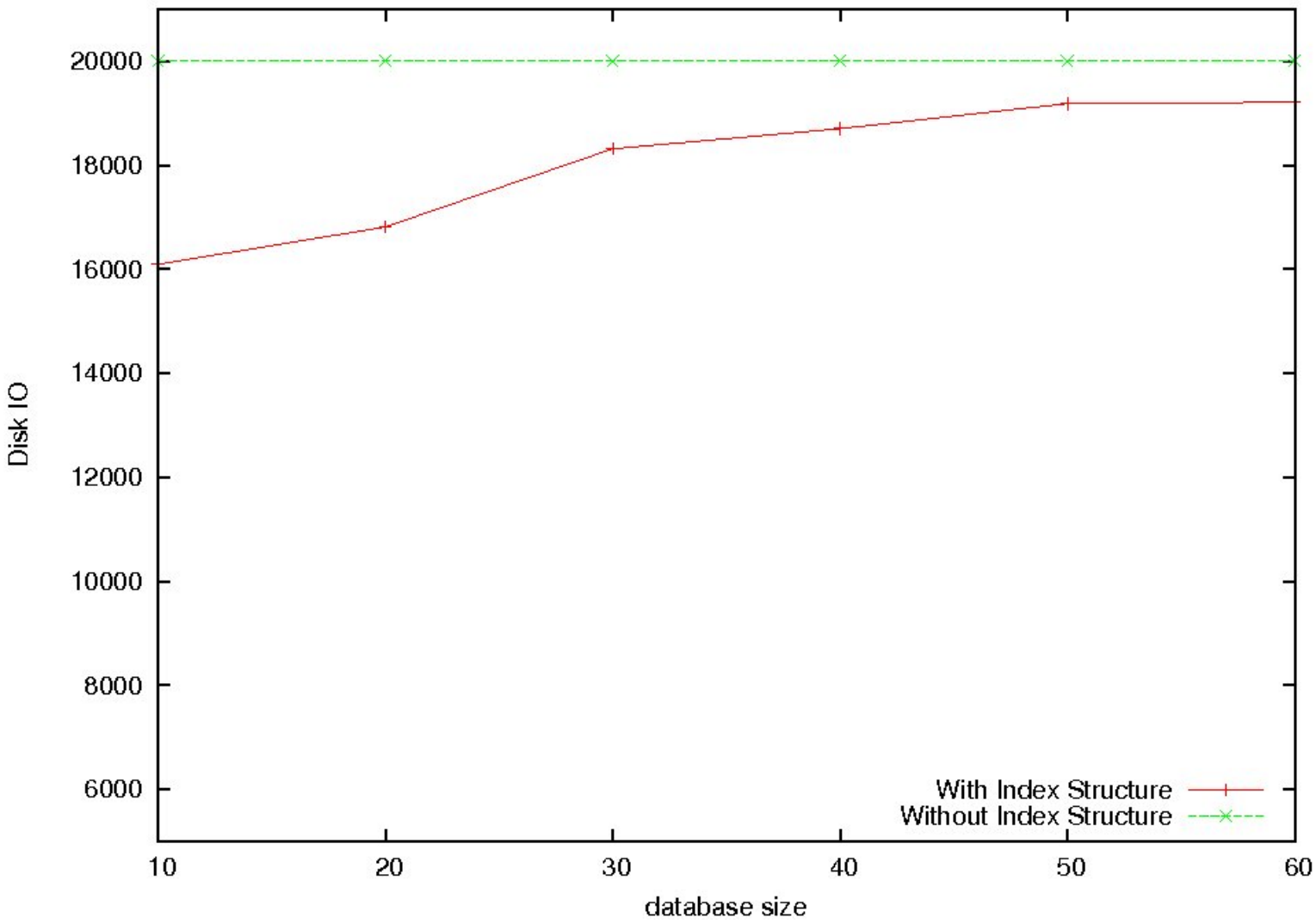
Results obtained



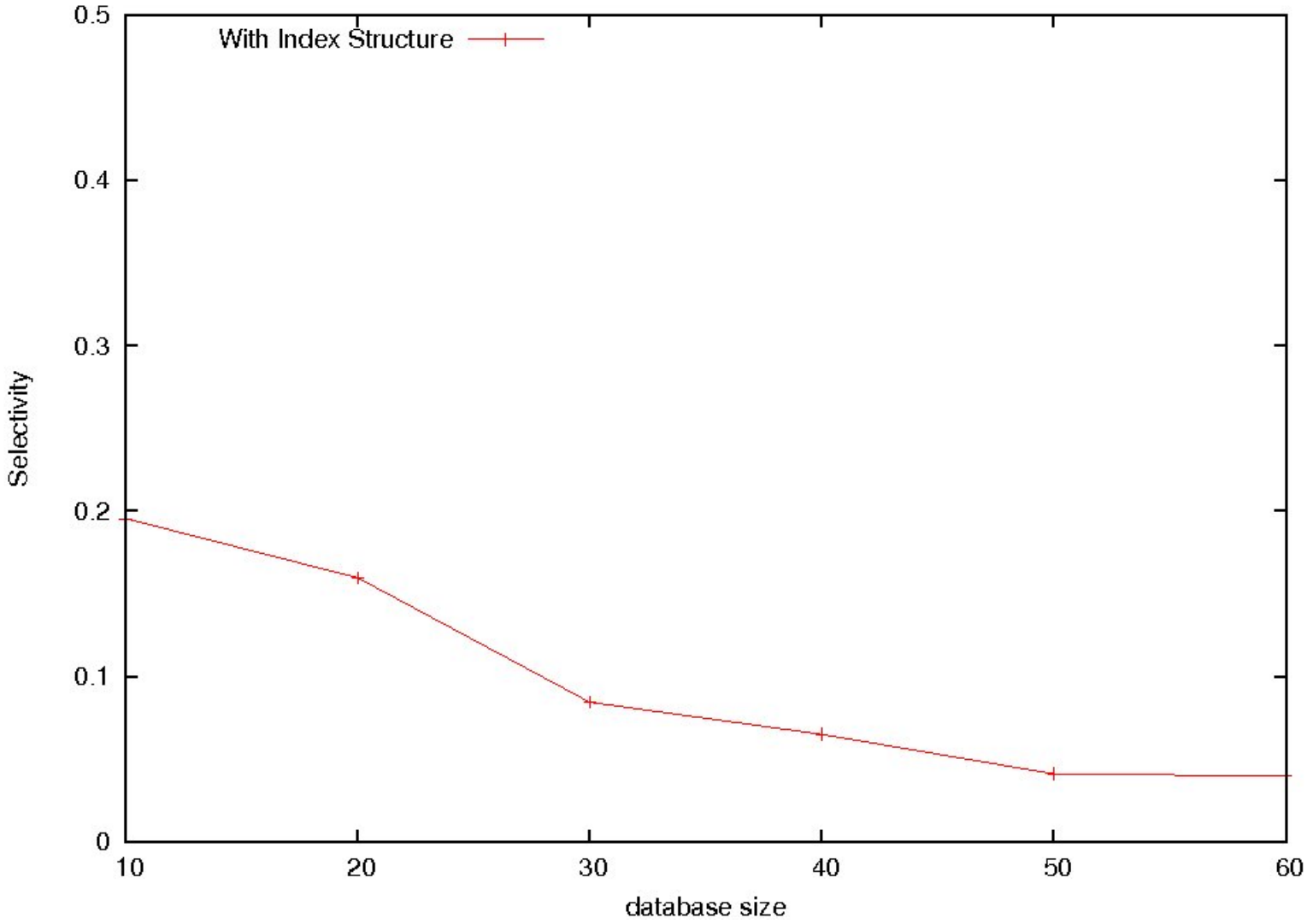
Results obtained



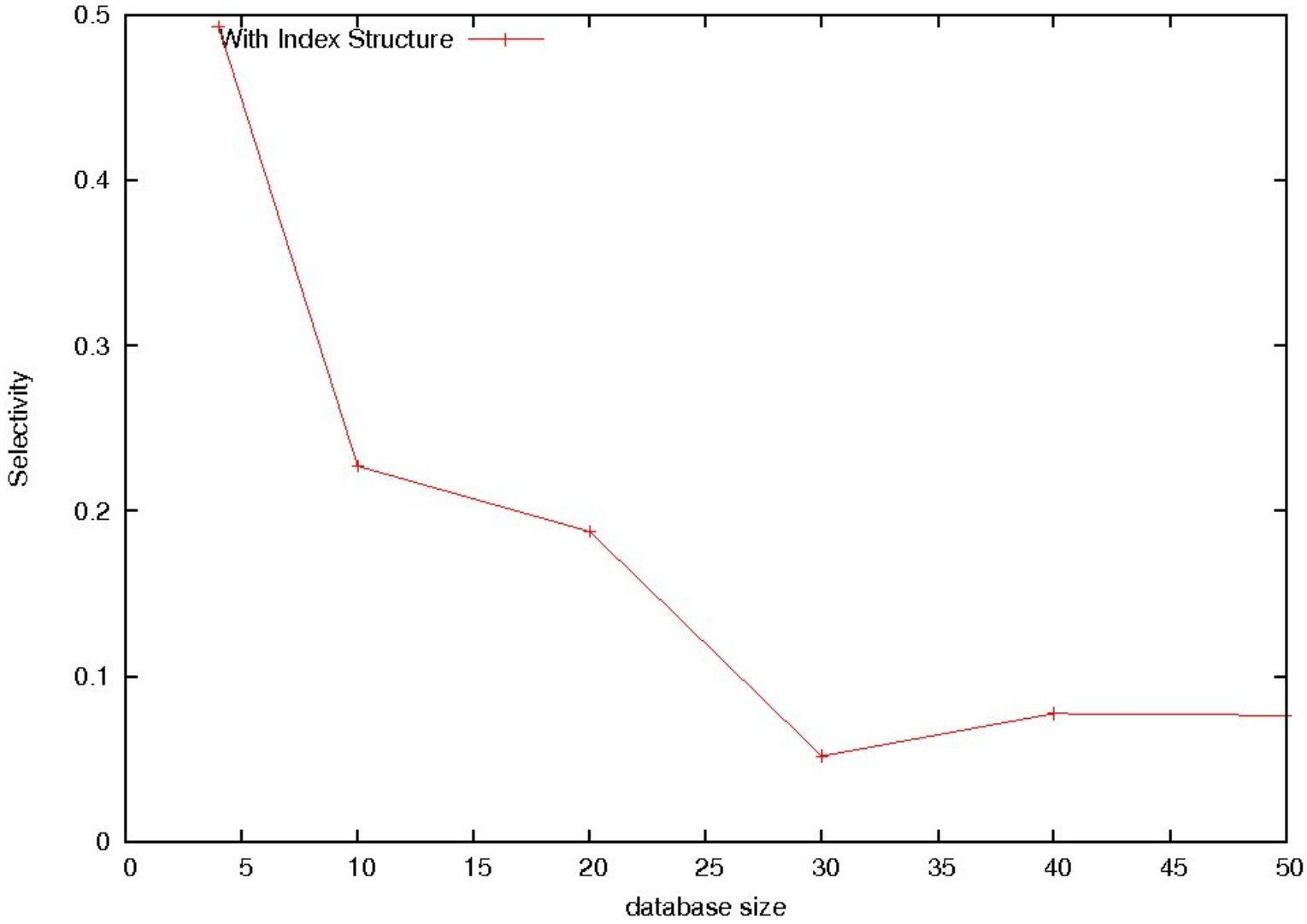
Results obtained



Results obtained



Results obtained



References

Nikolaus Augsten, Michael Böhlen, and Johann Gamper.
Approximate matching of hierarchical data using pq-grams.
In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 301–312. VLDB Endowment, 2005.

Jayendra Venkateswaran, Tamer Kahveci, Christopher M. Jermaine, and Deepak Lachwani.
Reference-based indexing for metric spaces with costly distance measures.
VLDB J., 17(5):1231–1251, 2008.