Indexing chemical fingerprints for efficient querying of molecular databases

Abhik Mondal (CS10B061)

IIT Madras

May 10, 2015

Project Guide: Dr. Sayan Ranu

Overview

1. Motivation
2. Problem Statement
   - M-tree based index
   - Inverted Index
3. Experiments and Results
4. Conclusion

Motivation

- Fast database search is vital in drug discovery, where the aim is identifying chemical compounds with high similarity to known drugs.

Motivation

- Fast database search is vital in drug discovery, where the aim is identifying chemical compounds with high similarity to known drugs.
- Why is a similarity search important?

Motivation

- Fast database search is vital in drug discovery, where the aim is identifying chemical compounds with high similarity to known drugs.
- Why is a similarity search important?

  ZINC database contains over 35 million purchasable compounds.

Motivation

- Fast database search is vital in drug discovery, where the aim is identifying chemical compounds with high similarity to known drugs.
- Why is a similarity search important?

  ZINC database contains over 35 million purchasable compounds.
- Exact Search?

Motivation

- Fast database search is vital in drug discovery, where the aim is identifying chemical compounds with high similarity to known drugs.
- Why is a similarity search important?

  ZINC database contains over 35 million purchasable compounds.
- Exact Search?

  Not looking for approximation methods. For example, Locally Sensitive Hashing.

Challenges

- Representation of molecules? Sub-graph Isomorphism is NP-complete. Solution?

Challenges

- Representation of molecules? Sub-graph Isomorphism is NP-complete. Solution?

  Fingerprint.

Challenges

- Representation of molecules? Sub-graph Isomorphism is NP-complete. Solution?

    Fingerprint.

- High dimensionality and sparseness of data.

Challenges

- Representation of molecules? Sub-graph Isomorphism is NP-complete. Solution?

  Fingerprint.

- High dimensionality and sparseness of data.

  Eg. Statistics of the PubChem dataset we have used:
  Number of data points : 264016
  Number of unique features : 785985
  Average number of features in a data point : 270.602966

Challenges

- Representation of molecules? Sub-graph Isomorphism is NP-complete. Solution?

  Fingerprint.

- High dimensionality and sparseness of data.

  Eg. Statistics of the PubChem dataset we have used:
  Number of data points : 264016
  Number of unique features : 785985
  Average number of features in a data point : 270.602966

- But why index?

Problem Statement

Range Search Problem

Given a fingerprint, say *'f'*, a similarity measure *'sim'*, a threshold distance $'\theta'$ and a database of chemical compounds D, we find the subset $S \subset D$ of all fingerprints, such that:

$$S = \{g \mid g \in D, sim(f, g) < \theta\} \tag{1}$$

Some more concepts/definitions

- Tanimoto similarity

$$T_s(X, Y) = \frac{\sum_i X_i \wedge Y_i}{\sum_i X_i \vee Y_i} \tag{2}$$

Some more concepts/definitions

- Tanimoto similarity

$$T_s(X, Y) = \frac{\sum\limits_i X_i \wedge Y_i}{\sum\limits_i X_i \vee Y_i} \tag{2}$$

- Min-Max similarity

$$M_s(X, Y) = \frac{\sum\limits_i min(X_i, Y_i)}{\sum\limits_i max(X_i, Y_i)} \tag{3}$$

Some more concepts/definitions

- Tanimoto similarity

$$T_s(X, Y) = \frac{\sum_i X_i \wedge Y_i}{\sum_i X_i \vee Y_i} \tag{2}$$

- Min-Max similarity

$$M_s(X, Y) = \frac{\sum_i min(X_i, Y_i)}{\sum_i max(X_i, Y_i)} \tag{3}$$

- Distance measure? Metric?

M-tree

- Routing objects
- Covering radius

M-tree

- Routing objects
- Covering radius

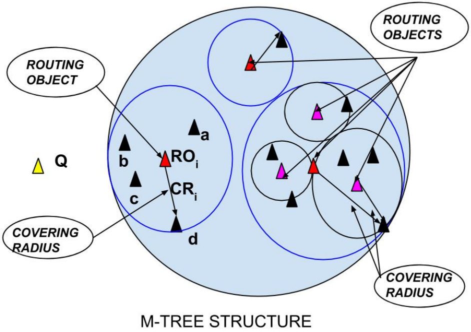


Figure : M-tree Structure Overview

Indexing approach ...

- Select pivots? Number?

Indexing approach ...

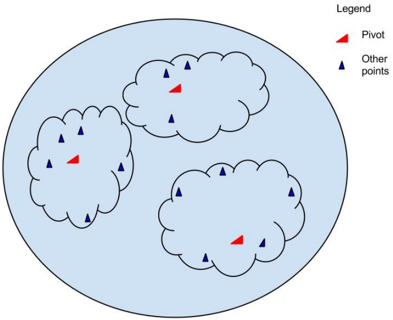
- Select pivots? Number?
  - The i^{th} pivot is chosen such that its minimum distance to the previous $i-1$ pivots is maximized.

Indexing approach ...

- Select pivots? Number?
  - The i^{th} pivot is chosen such that its minimum distance to the previous $i-1$ pivots is maximized.
- Assign each point to a pivot.

Indexing approach ...

- Select pivots? Number?
  - The i^{th} pivot is chosen such that its minimum distance to the previous $i-1$ pivots is maximized.
- Assign each point to a pivot.

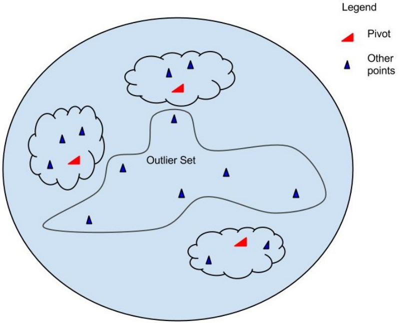


Indexing approach ...

- Choose outliers ?

Indexing approach ...

- Choose outliers ?

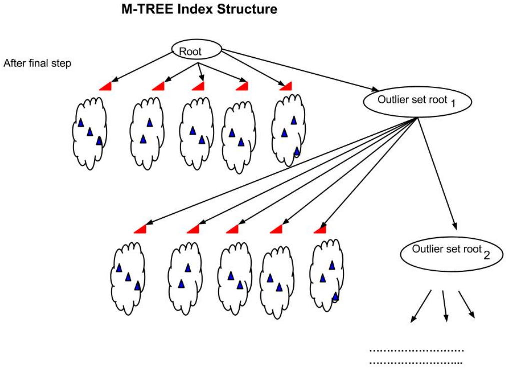


Indexing approach

- Repeat procedure on outlier set.

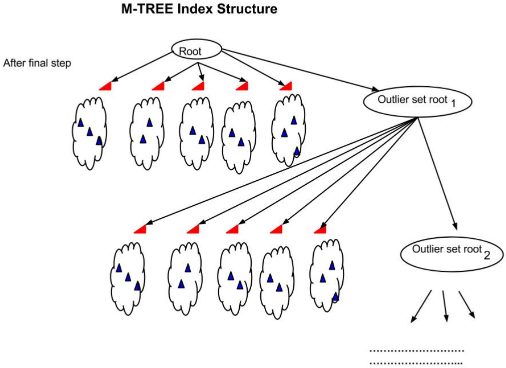
Indexing approach

- Repeat procedure on outlier set.



Indexing approach

- Repeat procedure on outlier set.



- Termination?

Range Search

- Start from the root as pivot p
- Apply triangle inequality bounds to prune or include all points from sub-tree.
- If not, then go to the children of p and repeat the process with them as the new pivot, till we reach leaf.

Inverted Index

- High dimensionality and sparsity of chemical data are an impediment to our indexing process.
- Use of inverted index motivated by its use in text mining.

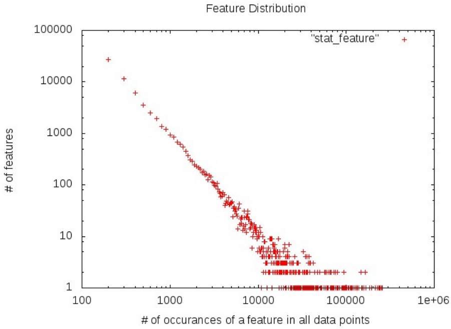
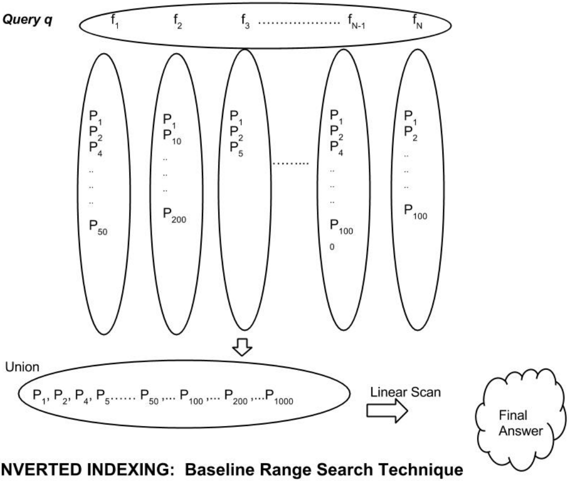


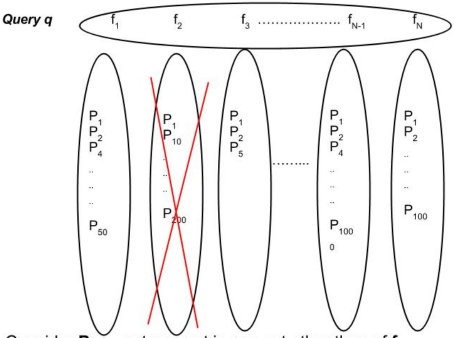
Figure : Distribution of data points against the features

Range Search



INVERTED INDEXING: Baseline Range Search Technique

Pruning



Consider $\mathbf{P_{200}}$, not present in any set other than of $\mathbf{f_2}$
Maximum Similarity possible for such a point with the query?

$$\mathbf{1 / (N_q - 1 + V_2)}$$

(N_q - number of features in query,
V_2 - minimum number of
features present in any point
containing f_2)

Can we prune the set ?

Greedy Technique

- Sort the features based on popularity

Greedy Technique

- Sort the features based on popularity
- Hence if till the i^{th} feature is considered, if j features (call it set R) have been pruned till now, we can prune the i^{th} feature as well if the following holds.

$$\frac{j+1}{N_q - 1 + min(V_i, \rho)} < 1 - t \tag{4}$$

where ρ is the minimum number of features present in any point containing atleast one of the features pruned until now i.e $\rho = \min_{k \in R} V_k$

Extension to non-binary fingerprints

Prune i^{th} feature if:

$$\frac{min(j_i, W_i)}{S_q - W_i - k_i + l_i + max(k_i, V_i)} < 1 - t \tag{5}$$

Here j_i is the maximum feature value taken for the feature f_i,
W_i is the i^{th} feature value of query q,
S_q is the sum magnitude of the feature values of the query q,
k_i is the is the minimum feature value taken for the feature f_i,
l_i is the minimum sum of feature values for any point containing the feature f_i ,
t is the threshold similarity.

Experiments

- Datasets
  - PubChem Dataset (264016 compounds, 785985 features)
  - DUD Dataset (128374 compounds, 32198 features)
- Evaluations
  - Compare range search result with that of full database scan
  - Compare average runtime of range search with the Bit bound technique [1]

[1] **Source:** Swamidass, S Joshua and Baldi, Pierre. *Bounds and algorithms for fast exact searches of chemical fingerprints in linear and sublinear time.*

M-tree based index analysis

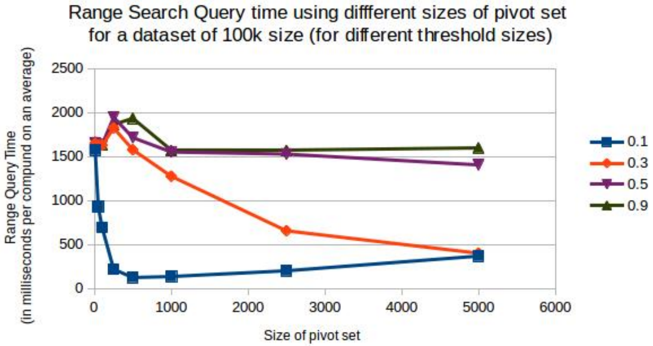
- Indexing time per compound on average increases linearly with data-set size as well as with size of pivot-set
- Outlier base limit size has no significant effect.

M-tree based index analysis...

- Runtime for different pivot-set sizes? High v/s low?

M-tree based index analysis...

- Runtime for different pivot-set sizes? High v/s low?



Inverted index analysis

- Indexing time per compound on average is constant. Does not change with data-set size.

Inverted index analysis

- Indexing time per compound on average is constant. Does not change with data-set size.
- Pruning upto 50-100 features on average for low threshold distances.

Comparison

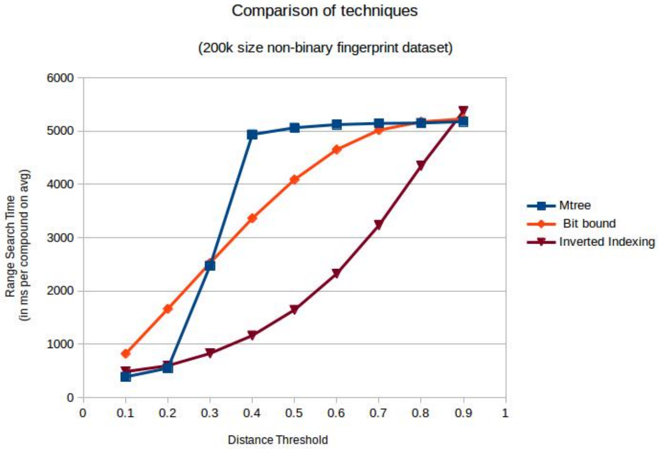


Figure : PubChem-n dataset

Comparison ...

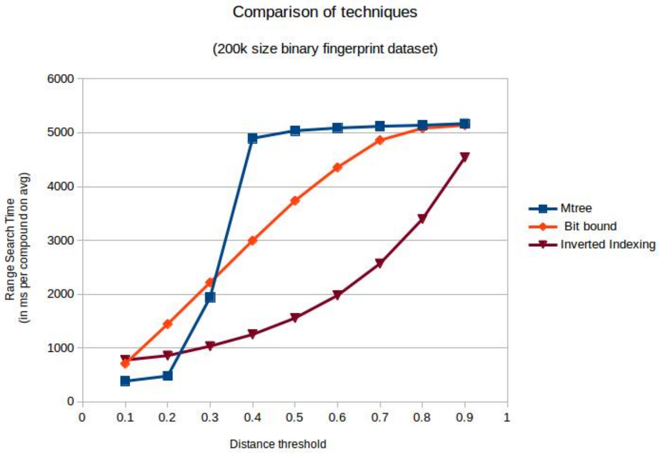


Figure : PubChem-b dataset

Comparison ...

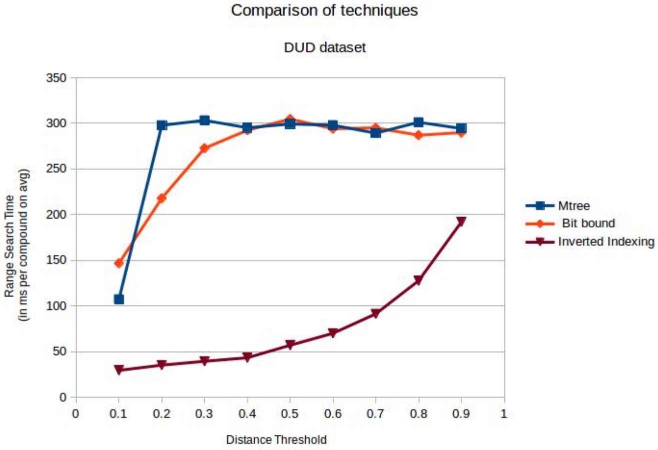


Figure : DUD dataset

Conclusion

- Proposed an M-tree based index approach which achieved 2-3 times speed-up over the Bit-Bound Technique.

Conclusion

- Proposed an M-tree based index approach which achieved 2-3 times speed-up over the Bit-Bound Technique.
- Proposed a novel inverted indexing technique which achieved 5-6 times speed-up over the Bit-Bound Technique.

Conclusion

- Proposed an M-tree based index approach which achieved 2-3 times speed-up over the Bit-Bound Technique.
- Proposed a novel inverted indexing technique which achieved 5-6 times speed-up over the Bit-Bound Technique.
- Showed the effectiveness of our techniques through comprehensive analysis on 2 real world datasets (both binary and non-binary).

Thank You!