AquaDiva
collaborative research center 1076

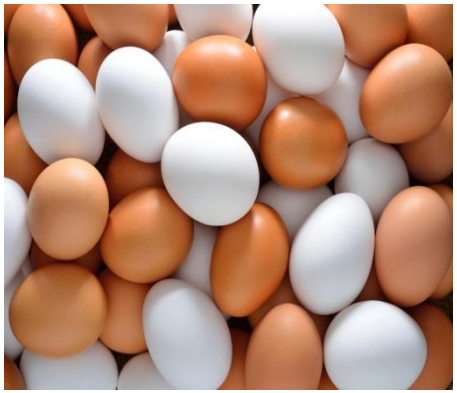
A Sequence-based Tree Similarity Search

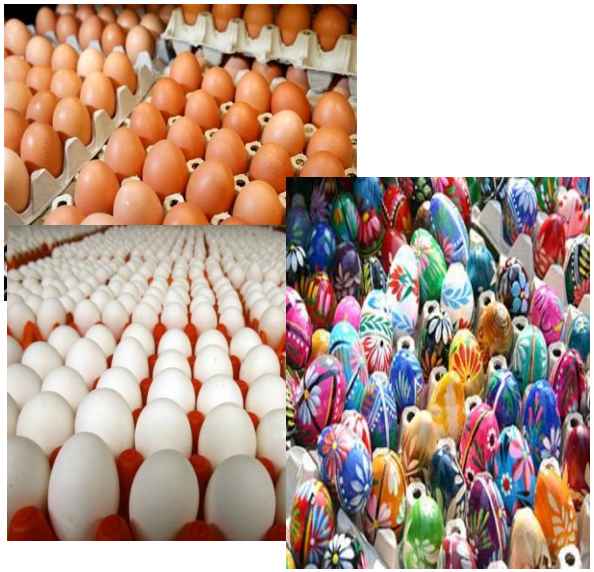
Alsayed Algergawy and Friederike Klan

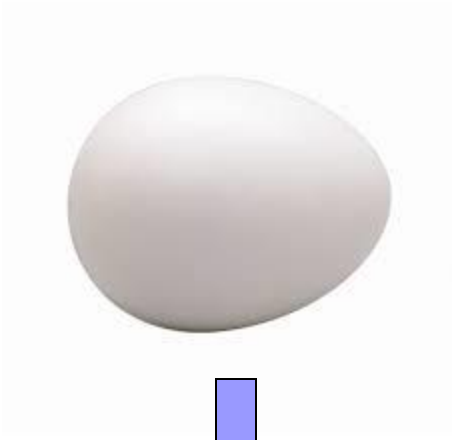
Heinz Nixdorf Chair for Distributed Information Systems

Friedrich Schiller University of Jena, Germany

Are these similar?







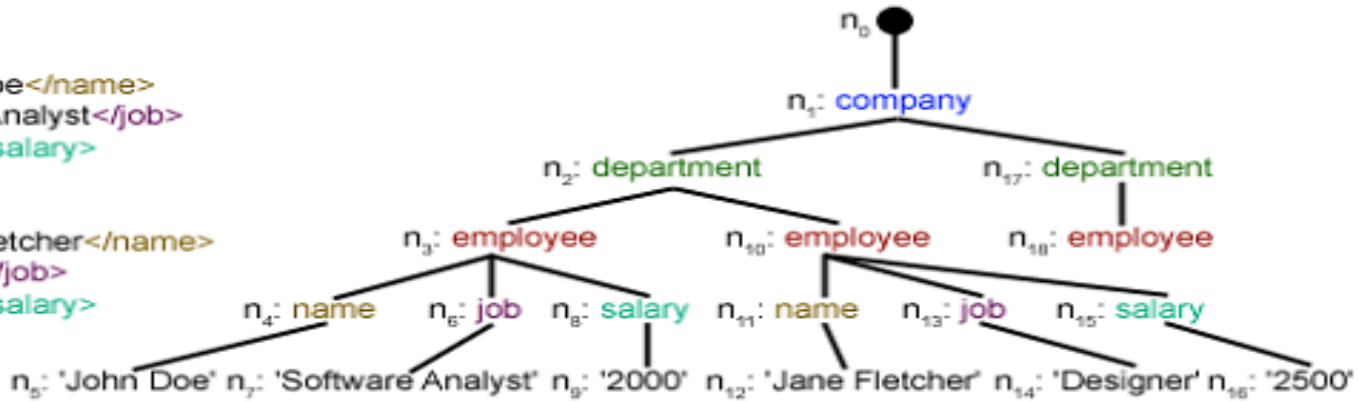
Outline

- Introduction
- Preliminary and sequence distance
- Proposed framework
- Preliminary results

Tree similarity search

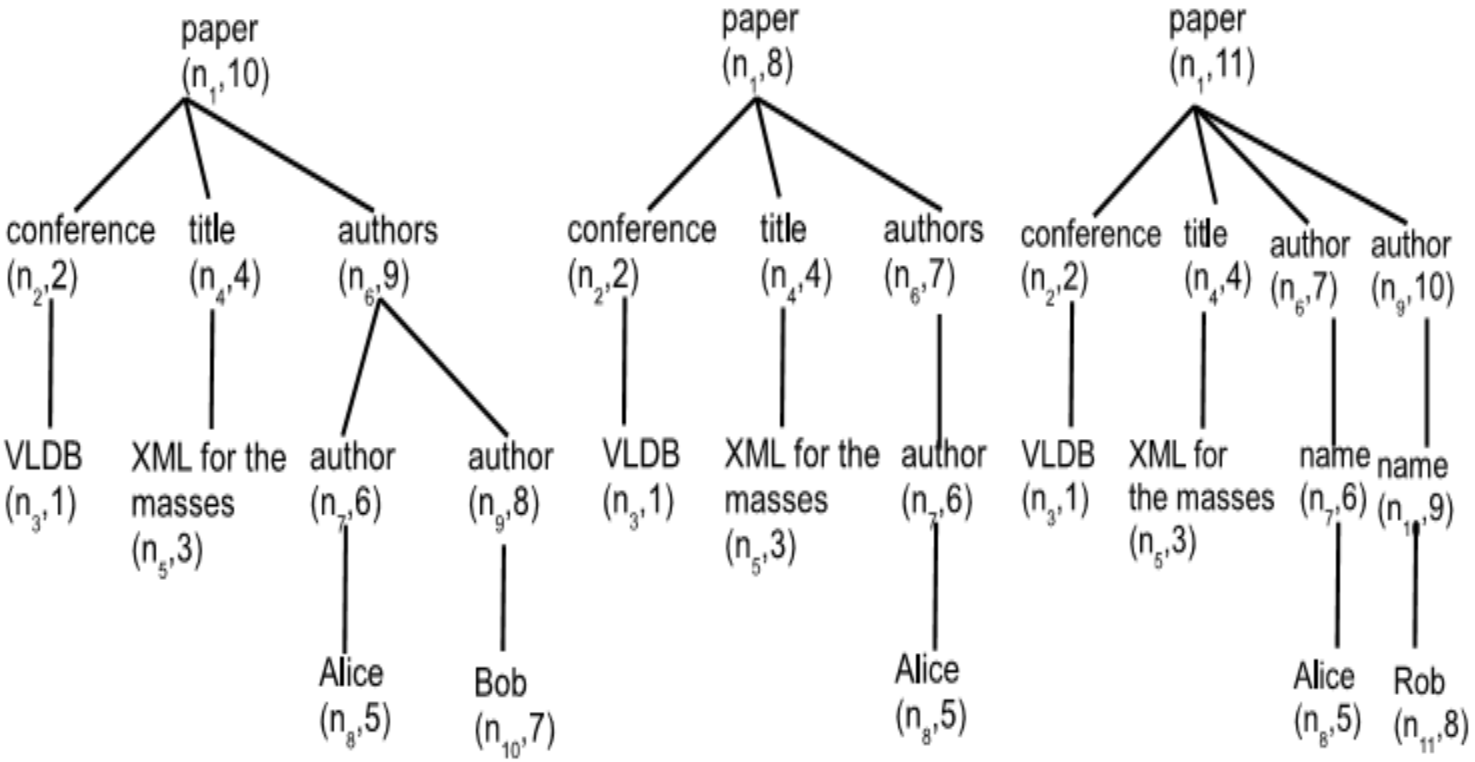
- What is similarity search?
  - The most general term used for a range of mechanisms which share the principle of searching (typically, very large) spaces of ***objects*** where the available comparator is the ***similarity*** between **any pair of objects**.
- Tree-structured data
  - hierarchical data is often modelled as ***trees***

```
<company>
 <department>
  <employee>
   <name>John Doe</name>
   <job>Software Analyst</job>
   <salary>2000</salary>
  </employee>
  <employee>
   <name>Jane Fletcher</name>
   <job>Designer</job>
   <salary>2500</salary>
  </employee>
 </department>
 <department>
  <employee>
  </employee>
 </department>
</company>
```



Tree similarity search

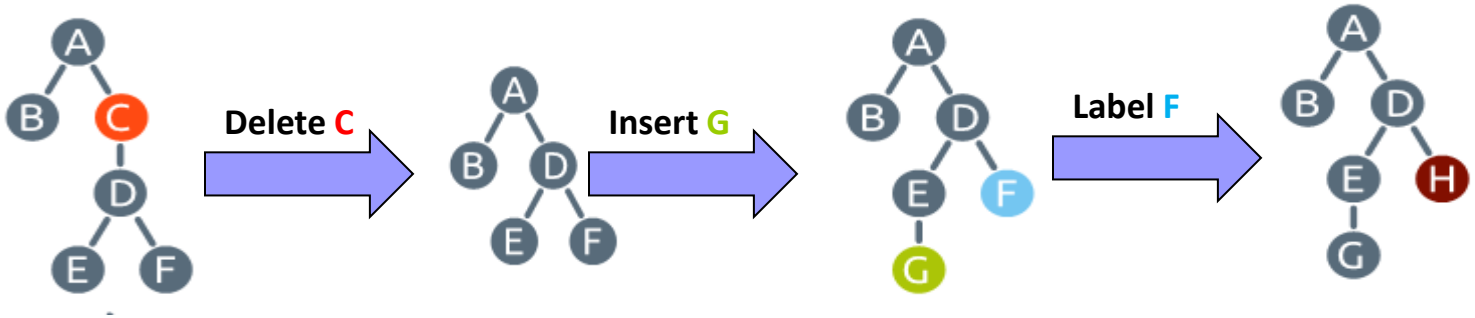
- Problem definition
  - *Tree structured data* getting more and more important for many scientific and modern databases applications



Tree Edit Distance (TED)

- TED is a standard measure to tree similarity
- TED is the minimum cost sequence of edit operations required to transform T_1 to T_2

$$TED(T_1, T_2) = \sum_{i=1}^{k} c_i * ei(n)$$



Dynamic programming solution to TED

- TED has a recursive solution which decomposes trees into sub-forests
- use distances of smaller sub-problems to compute larger

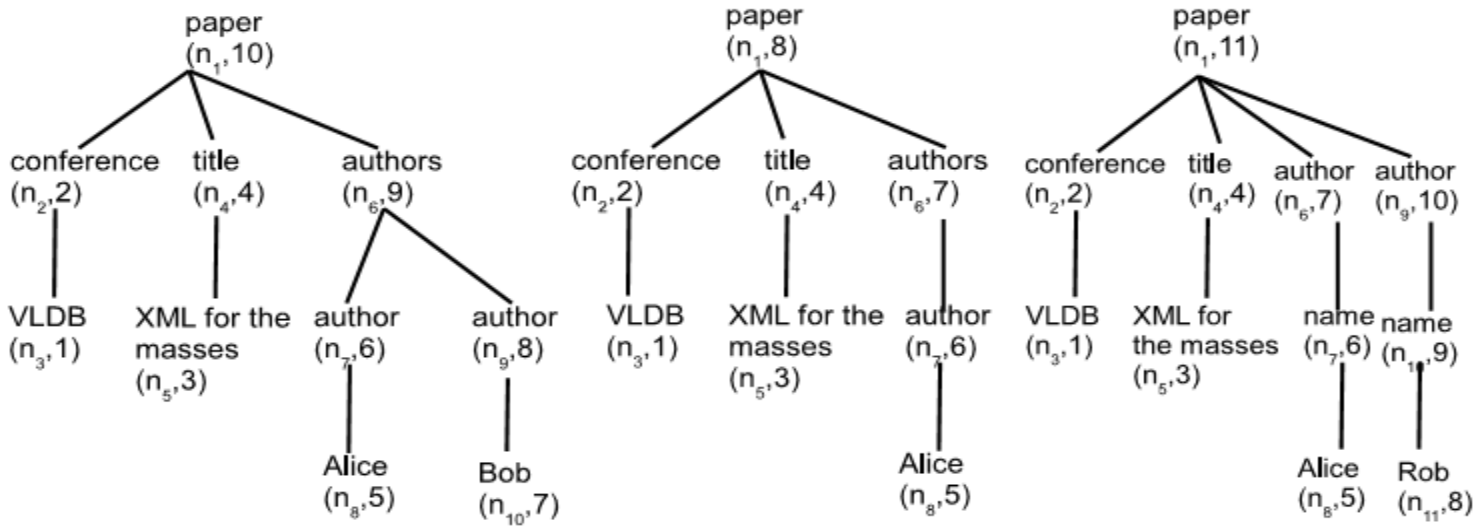
Algorithm	Time	Space
Tai, 1979	$O(n^6)$	$O(n^6)$
Zhang & Shasha 89	$O(n^4)$	$O(n^2)$
Klein, 98	$O(n^3 \log n)$	$O(n^3 \log n)$
Demaine at el., 2009	$O(n^3)$	$O(n^2)$
Pawlik & Augsten, 2011	$O(n^3)$	$O(n^2)$

Prufer sequence representation

- In our implementation,

  $T = (N, E, LabN)$; *distinguishing between two types of nodes*

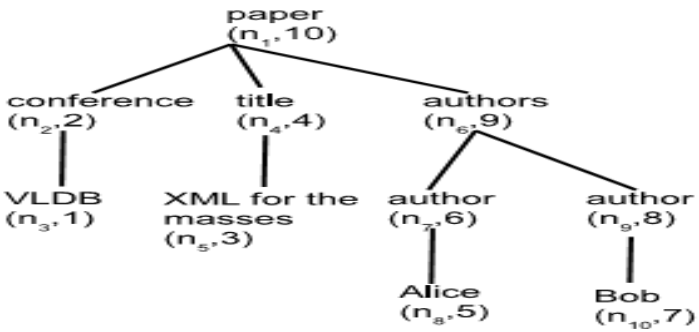
  - Element node (internal)
  - Value node (leaf)



$NPS(T_a) =$	(2	10	4	10	6	9	8	9	10	-)
$LPS(T_a).OID =$	(n_3	n_2	n_5	n_4	n_8	n_7	n_{10}	n_9	n_6	n_1)
$LPS(T_a).label =$	(VLDB	conference	XML for the masses	title	Alice	author	Bob	author	authors	paper)

CPS properties

- $CPS(T) = (NPS, LPS)$; *and* $n_i \in T$ *with post* k *distinguishing between two types of nodes*
  - $if\ k \in NPS$, then n_i is an element node (internal)
  - $if\ k \in NPS$, then n_i is a value node (leaf)
  - $if\ k \in NPS$, appears m times, then n_i has exactly m children
  - The most important property of the sequence representation is that *NPS* is a unique representation of the data tree.



$NPS(T_a) =$	(2	10	4	10	6	9	8	9	10	-)
$LPS(T_a).OID =$	$(n_3$	n_2	n_5	n_4	n_8	n_7	n_{10}	n_9	n_6	$n_1)$
$LPS(T_a).label =$	(VLDB	conference	XML for the masses	title	Alice	author	Bob	author	authors	paper)

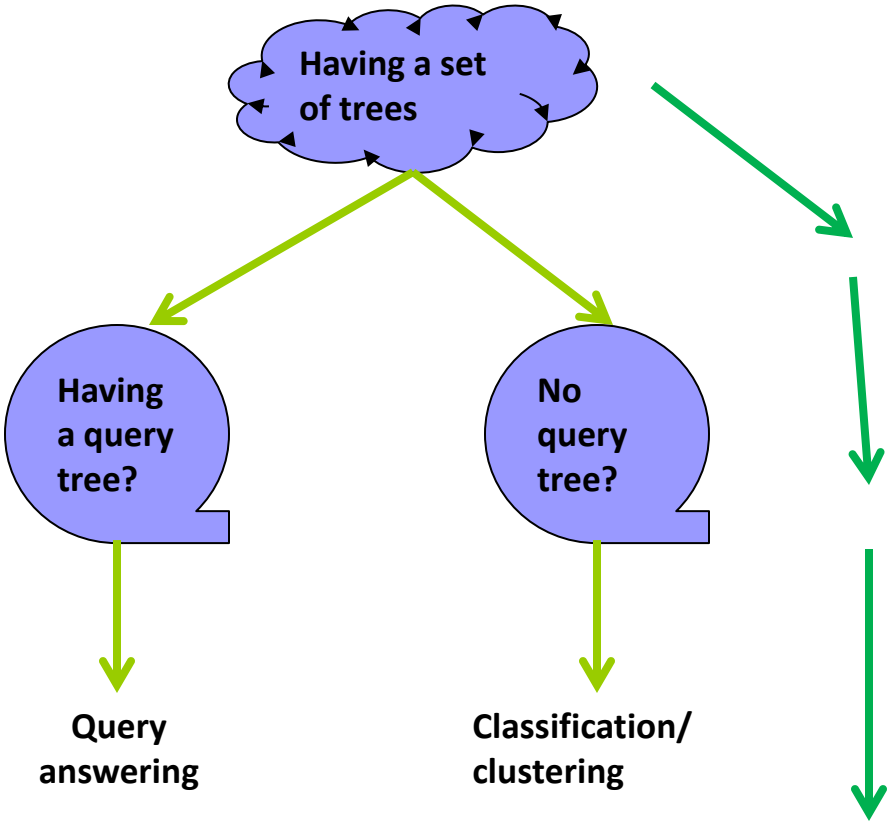
Tree signature distance (TSD)

- $NPS(T_1) = (n_1, n_2, \ldots, nk_1)$; *and* $NPS(T_2) = (\hat{n}_1, \hat{n}_2, \ldots, \hat{n}_{k2})$, we define a new tree distance, called Tree sequence distance

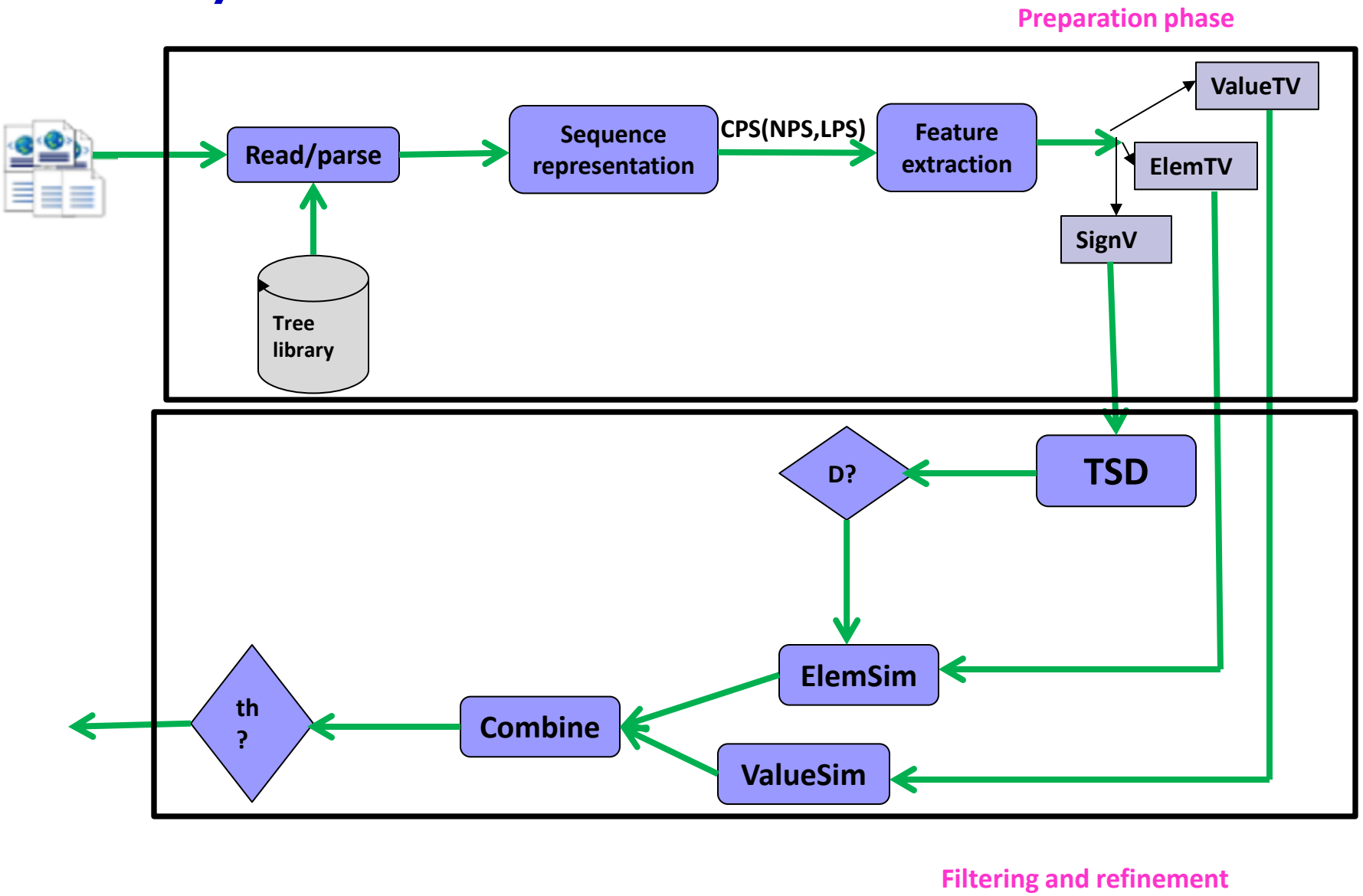
$$TSD(T_1, T_2) = \frac{\sum_{i=1}^{\max(k_1,k_2)} |s_i - \hat{s}_i|}{\min(k_1, k_2)}$$

- Having the following properties:
  - $TSD(T_1, T_2) \geq 0$, and $TSD(T_1, T_1) = 0$
  - $TSD(T_1, T_2) = TSD(T_2, T_1)$
  - $TSD(T_1, T_3) \leq TSD(T_1, T_2) + TSD(T_2, T_3)$

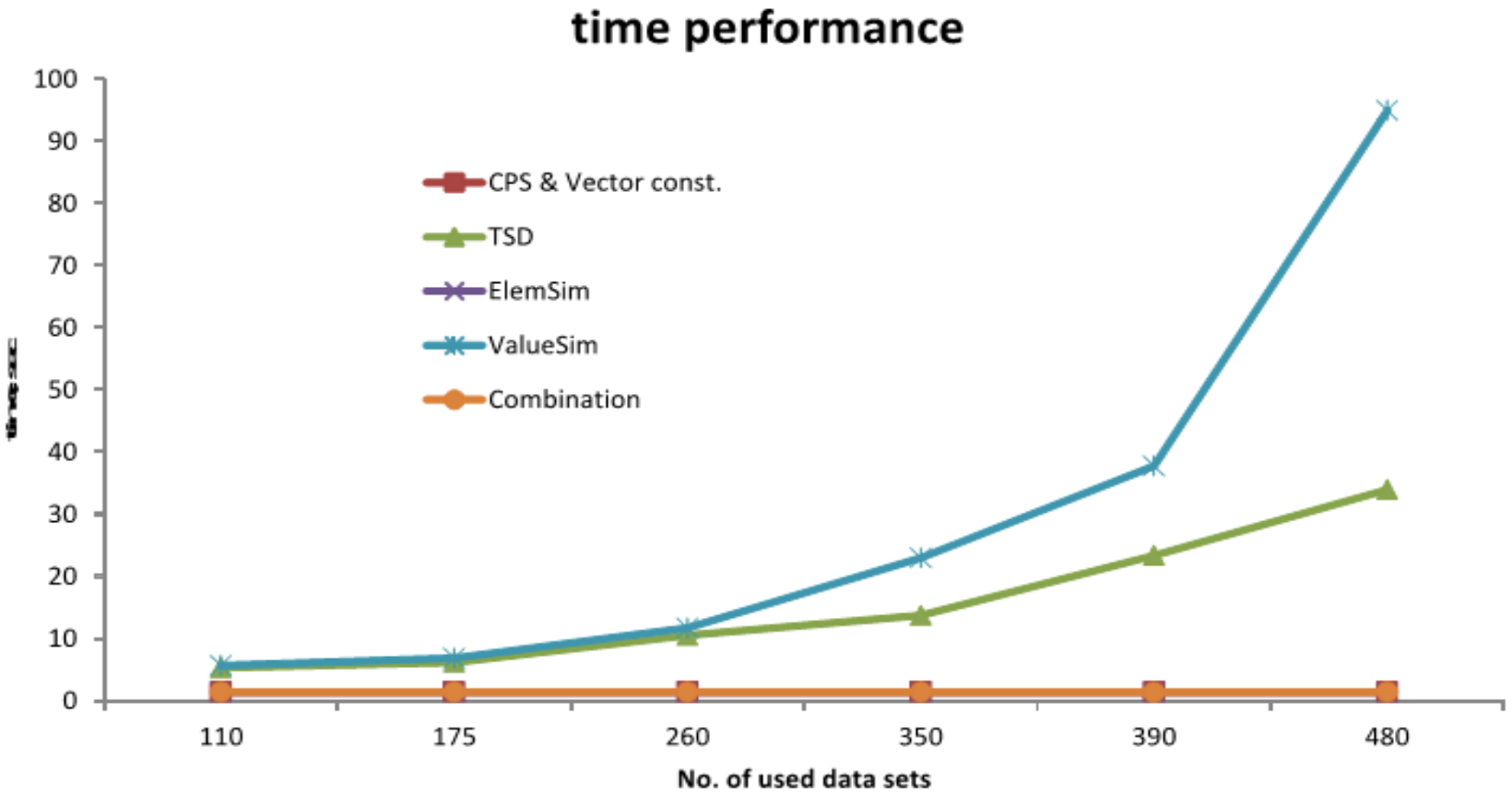
Similarity search framework



Similarity search framework



Experimental evaluation: preliminary results



Sum up

- Addressing the problem of tree similarity search;
  - Focusing on classification and clustering cases
- Introducing tree signature
  - Based on this representation, introducing a tree sequence distance
- Conducting a set of experiments
  - Results are accurate and fast
- Ongoing work
  - Extending the framework to be used for query processing
  - Comparing with other approaches

Thanks