CaseCluster: Visualization of Supreme Court Cases

Georg Apitz and Neeti Ogale
University of Maryland
{geapi,neeti}@cs.umd.edu
Problem Description

- Broad problem: Visualization of sets

  1 – [a,b,d,e]  
  2 – [d,e,f,z]  
  3 – [a,b,f,k,z]  
  4 – [a,b,c,d]  
  5 – [d,f,x]

- Our focus is on the specific problem of visualizing sets of Supreme Court case references
  
  Full dataset: 2561 cases and total 28,953 references
Clustering

- Make groups of elements with
  - High intra-group similarity
  - High inter-group dissimilarity.
  - Similarity is a measure of the intersection set of the references.

- Our clustering produces a tree in which each level has disjoint clusters.
Demo
Algorithm

- Recursive top-down algorithm.
- Start at the root and descend, splitting the clusters at each level
- Input from the user: MaximumTreeDepth, MinimumClusterSize
<table>
<thead>
<tr>
<th>Websites</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.aol.com">www.aol.com</a></td>
<td>Arts, Entertainment, Humanities, Autos, Business, Computers, Culture, Health, Home,</td>
</tr>
<tr>
<td><a href="http://www.yahoo.com">www.yahoo.com</a></td>
<td>Humanities, Business, Government, Kids, Teens, Science, Society, Culture,</td>
</tr>
<tr>
<td><a href="http://www.shopping.com">www.shopping.com</a></td>
<td>Computers, Garden, Beauty, Home, Clothing, Sports</td>
</tr>
</tbody>
</table>
Algorithm in steps

- Build a clustering tree, starting at the root. The root contains all the elements '{Common set: 0}': AOL, Yahoo, Ebay, Dmoz, Shopping
Step 2: Recursively split the clusters

2) Find the element which is referenced by maximum elements in the group

AOL, Yahoo, Ebay, Dmoz, Shopping
{Common set: 0}

AOL, Yahoo, Ebay
Common set:(1): Business

AOL, Yahoo
Common set:(2): Business, Humanities

Shopping, Dmoz
{Common set: 1}: Computers

Ebay

3) If the cases in a cluster go below the MinimumClusterSize, group all remaining cases in a collector node and this node is not split further.

Common set:(2): Business, Kids
Key points of the algorithm

- Builds a k-ary tree, in which each level has disjoint clusters.
- Clusters at level D have D common references.
- Runtime of the algorithm is $O(n \log n)$ for n cases.
- Maximum referenced case is given priority in the clustering.
Advantages

- Fast algorithm allows **interactivity**.
  - For our full dataset it takes about 1 sec.

- Makes no assumptions about the input data.
  - If the data has no clustering pattern the cases will be grouped in the collector node
Future work

- Extend algorithm to create clusters based on containment and resemblance.
- The system can be extended to have other visualizations of the hierarchical data.
- Other metrics such as probabilities of co-citations can be shown in the co-citation matrix.
Thank you!