Transforming XML trees for efficient classification and clustering (structure only task)

L. Candillier\textsuperscript{1,2}, I. Tellier\textsuperscript{1}, F. Torre\textsuperscript{1}
\textsuperscript{1}GRAppA, Lille 3 University / \textsuperscript{2}Pertinence, Paris

1 Tackling XML trees
   • Existing methods
   • Trees transformations

2 Subspace clustering for XML
   • Algorithm SSC
   • Adaptations of SSC for XML

3 Experiments
   • Boosted C5 on transformed XML documents
   • Adaptations of SSC for XML

4 Conclusion
Tackling XML trees

Existing methods

I. Tackling XML trees

Work directly on the trees

- Based on the edit distance
  [Nierman and Jagadish, 2002, Dalamagas et al., 2004]
- Based on the number of common paths
  [Flesca et al., 2002, Lian et al., 2004, Costa et al., 2004]
- Discovering frequent subtrees
  [Termier et al., 2002, Zaki and Aggarwal, 2003]

Use different kind of representation for their manipulation

- Bag-of-tags [Doucet and Ahonen-Myka, 2002]
- Richer representation
Trees transformations

Possible transformations

- tags & occurrences
- parent-child relations & occurrences
- next-sibling relations & occurrences
- distinct paths starting from the root & occurrences
- node positions & number of children

⇒ need an algorithm able to handle many attributes, and to perform feature selection during the learning process

- boosted C5 [Quinlan, 2004] on the entire set of attributes
- subspace clustering [Parsons et al., 2004] considering different levels in the sets of attributes
II. Subspace clustering for XML

Different clusters may exist in different subspaces

\[ \in X \times Z \]
\[ \in X \times Z \]
\[ \in Y \times Z \]
\[ \in X \times Y \]
Algorithm SSC [Candillier et al., 2005]

- Based on the use of probabilistic models
- Assumption of independent distributions on each dimension
- EM algorithm [Ye and Spetsakis, 2003]
- Understandable output presentation

Adaptations for XML

- unsupervised version
- supervised version
Summary of SSC algorithm

Given $K$ the number of expected clusters

1. **Clusters detection**
   - Iterate $R$ times
     - Initialize the model (randomly)
     - Optimize the model parameters ($EM$)
     - Compute the log-likelihood $LL$
   - Select the model that maximizes $LL$

2. **Output presentation**
   - Create the rules associated with the clusters
   - Simplify the rules
EM algorithm

Find the model parameters that best fit the data
⇒ optimize the log-likelihood $LL$ of the model to the data

Iterate 2 steps

1. **Expectation**: find the membership probabilities of the data to the clusters according to the current model parameters

2. **Maximization**: update the model parameters according to the new membership probabilities

Stop when $LL^{t+1} - LL^t < \delta$
As a set of rules (hypercubes)

\[ \Rightarrow \text{associate to each dimension the smallest interval containing all the data points of the cluster} \]
Output presentation

As a set of rules (hypercubes)

⇒ associate to each dimension the smallest interval containing all the data points of the cluster

+ select as few dimensions as possible
## Feature selection

1. \( W_{kd} = \) weight of dimension \( d \) for cluster \( C_k \)
   
   \( = \) ratio between local and global standard deviation

2. Select the \( nb_{ds} \) dimensions of higher weights for each cluster
   (user parameter)

3. Delete, in ascending order of their weights, the dimensions from the rule if their deletion does not modify its support
Adaptations of SSC for XML

\[ A = \text{set of possible attributes associated to the trees} \]

- \( A_1 = \text{set of tags} \)
- \( A_2 = \text{set of parent-child relations} \)
- \( A_3 = \text{set of next-sibling relations} \)
- \( A_4 = \text{set of node positions} \)
- \( A_5 = \text{set of paths starting from the root} \)
while the number of clusters is not the one expected
- for each current cluster $C_k$ and for each set of possible attributes $A_i$, compute the interest of partitioning $C_k$ into 2 parts according to $A_i$
- select the best cut, compute the corresponding rule, and use this rule as the next test in the output decision tree

interest = ratio between likelihood of the model for 2 clusters and likelihood of the model for 1 cluster, weighted by the number of data points in the cluster

output = decision tree where each node corresponds to a membership test to a rule
Adaptation for understandable classification

1. **clustering preserving the classes**: allows to mix various classes in one cluster but does not allow a class to be splitted into different clusters

2. **separate the classes still embedded in the same clusters**
   - use rules as long as possible (more understandable)
   - else, use the best probabilistic model (cross-validation error)

3. **output = decision tree** where a node can correspond to membership tests to rules, or to probability tests on probabilistic models
III. Experiments

Structure only tasks

- **C5 boosted** 10 times on transformed datasets inex-s, m-db-s-0, m-db-s-1, m-db-s-2 and m-db-s-3

- **Adaptations of SSC** for XML on m-db-s-0 only \((nb\_ds = 10)\)
Boosted C5 on transformed XML documents

Number of attributes generated for each dataset

<table>
<thead>
<tr>
<th>dataset</th>
<th>nb of tags</th>
<th>nb of parent-child relations</th>
<th>nb of next-sibling relations</th>
<th>nb of node positions</th>
<th>nb of paths</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>inex-s</td>
<td>150</td>
<td>1038</td>
<td>827</td>
<td>2475</td>
<td>3674</td>
<td>8164</td>
</tr>
<tr>
<td>mdbso</td>
<td>197</td>
<td>2172</td>
<td>419</td>
<td>6575</td>
<td>320</td>
<td>9683</td>
</tr>
<tr>
<td>mdbsi</td>
<td>197</td>
<td>6477</td>
<td>5617</td>
<td>9159</td>
<td>16772</td>
<td>38222</td>
</tr>
<tr>
<td>mdbso</td>
<td>196</td>
<td>8953</td>
<td>7455</td>
<td>9183</td>
<td>25628</td>
<td>51415</td>
</tr>
<tr>
<td>mdbso</td>
<td>199</td>
<td>10639</td>
<td>9557</td>
<td>8537</td>
<td>37576</td>
<td>66508</td>
</tr>
</tbody>
</table>
Error rates of boosted C5 on the datasets transformed into attribute-values

<table>
<thead>
<tr>
<th>dataset</th>
<th>error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>inex-s</td>
<td>0.011</td>
</tr>
<tr>
<td>m-db-s-0</td>
<td>0.026</td>
</tr>
<tr>
<td>m-db-s-1</td>
<td>0.038</td>
</tr>
<tr>
<td>m-db-s-2</td>
<td>0.062</td>
</tr>
<tr>
<td>m-db-s-3</td>
<td>0.062</td>
</tr>
</tbody>
</table>
Tree obtained when clustering dataset m-db-s-0

Tag(title) = 0 & Tag(BD) ≤ 2

Parent(AW – BC) = 1

Tag(BQ) = 1

Next – sibling(CU – CV) = 1

R4

R7

R10

R6

R11
Tree obtained for classifying dataset m-db-s-0 (error=0.03)

Tag(movie) = 1

1

Tag(CL) = 1

S2 > S3

2

3

Tag(BG) = 1

S4 > S5

4

5

Tag(BA) = 0

Parent(AT - AQ) = 0

7

9

8

Nb(0.0.0) = 0

P6 > P11

10

6

11

Tag(AJ) = 1
Conclusion

Contribution

- Method for transforming trees into sets of attribute-values
  - Need then to use methods able to handle many attributes and to perform feature selection during the learning process
  - Allows us to benefit from the strengths of existing methods
- New methods providing interpretable outputs

Future work

- Use other representation: forks / localisation of the relations
- Find a compromise between the number of new created attributes and the information they carry


In 3rd Hellenic Conference on Artificial Intelligence, Samos, Greece.


IEEE transactions on Knowledge and Data Engineering, 16(1) :82–96.

Evaluating structural similarity in XML documents.
In 5th International Workshop on the Web and Databases (WebDB 2002), Madison, Wisconsin, USA.

Evaluating subspace clustering algorithms.
In Workshop on Clustering High Dimensional Data and its Applications, SIAM Int. Conf. on Data Mining, pages 48–56.

Data mining tools see5 and c5.0.

Treefinder : a first step towards xml data mining.
In IEEE International Conference on Data Mining (ICDM02), pages 450–457.