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

L. Candillier[1,2], I. Tellier[1], F. Torre[1]
[1]GRAppA, Lille 3 University / [2]Pertinence, Paris

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

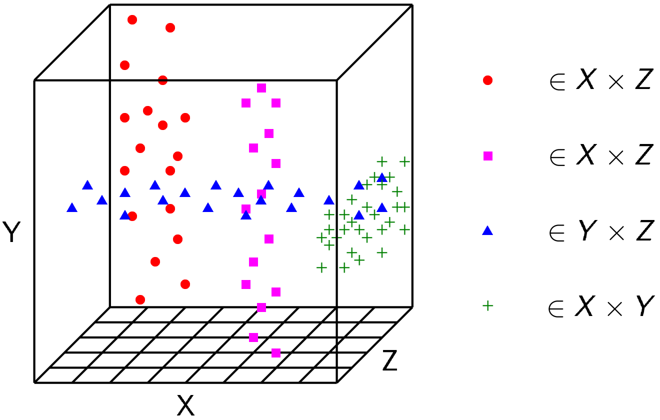
- tags & occurences
- parent-child relations & occurences
- next-sibling relations & occurences
- distinct paths starting from the root & occurences
- node positions & number of children

$\Rightarrow$ 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



Statistical Subspace Clustering

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
$\Rightarrow$ optimize the *log-likelihood LL* of the model to the data

Iterate 2 steps

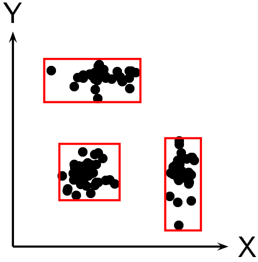
1. Expectation : find the membership probabilities of the data to the clusters according to the current model parameters
2. Maximization : update the model parameters according to the new membership probabilities

Stop when $LL^{t+1} - LL^{t} < \delta$

Output presentation

As a set of rules (hypercubes)

$\Rightarrow$ associate to each dimension the smallest interval containing all the data points of the cluster

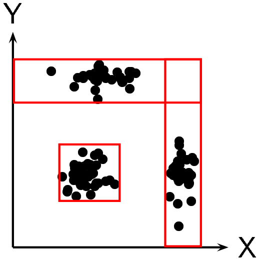


Output presentation

As a set of rules (hypercubes)

$\Rightarrow$ 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 = set of possible attributes associated to the trees

- A_1 = set of tags
- A_2 = set of parent-child relations
- A_3 = set of next-sibling relations
- A_4 = set of node positions
- A_5 = set of paths starting from the root

SSC for XML

- 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 partitionning 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
    1. use rules as long as possible (more understandable)
    2. 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 ($nb\_ds = 10$)

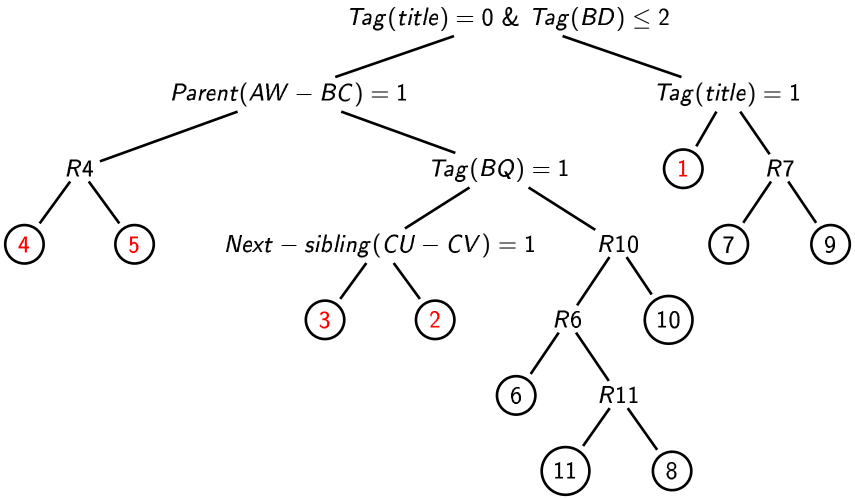
Number of attributes generated for each dataset

dataset	nb of tags	nb of parent-child relations	nb of next-sibling relations	nb of node positions	nb of paths	total
inex-s	150	1038	827	2475	3674	8164
mdbs0	197	2172	419	6575	320	9683
mdbs1	197	6477	5617	9159	16772	38222
mdbs2	196	8953	7455	9183	25628	51415
mdbs3	199	10639	9557	8537	37576	66508

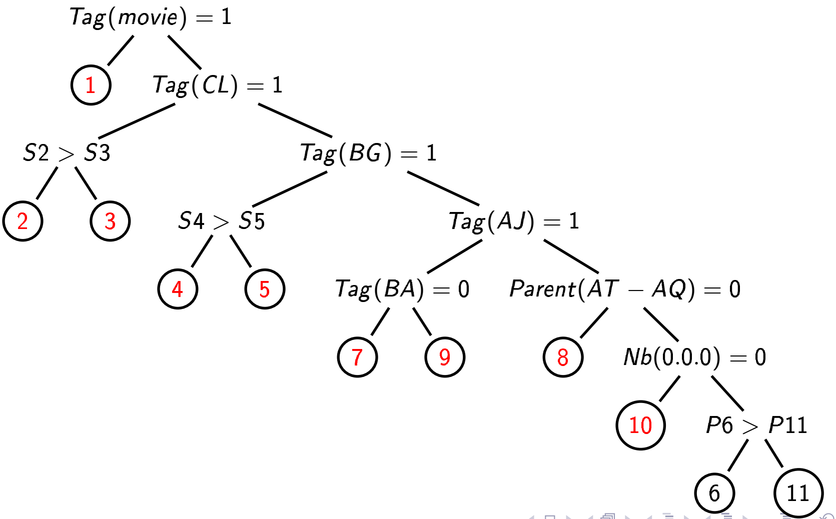
Error rates of boosted C5 on the datasets transformed into attribute-values

dataset	error rate
inex-s	0.011
m-db-s-0	0.026
m-db-s-1	0.038
m-db-s-2	0.062
m-db-s-3	0.062

Tree obtained when clustering dataset m-db-s-0



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



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

Candillier, L., Tellier, I., Torre, F., and Bousquet, O. (2005).
SSC : Statistical Subspace Clustering.
In Perner, P. and Imiya, A., editors, 4th International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM'2005), volume LNAI 3587 of LNCS, pages 100–109, Leipzig, Germany. Springer Verlag.

Costa, G., Manco, G., Ortale, R., and Tagarelli, A. (April 2004).
A tree-based approach to clustering XML documents by structure.
Technical report, Institute of Italian National Research Council, Rende, Italy.

Dalamagas, T., Cheng, T., Winkel, K.-J., and Sellis, T. (May 2004).
Clustering XML documents by structure.

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

Doucet, A. and Ahonen-Myka, H. (2002).
Naïve clustering of a large XML document collection.
In 1st Annual Workshop of the Initiative for the Evaluation of XML retrieval (INEX'02), Schloss Dagstuhl, Germany.

Flesca, S., Manco, G., Masciari, E., Pontieri, L., and Pugliese, A. (2002).
Detecting structural similarities between XML documents.
In 5th International Workshop on The Web and Databases (WebDB'02), Madison, Wisconsin.

Lian, W., Cheung, D. W., Mamoulis, N., and Yiu, S.-M. (January 2004).
An efficient and scalable algorithm for clustering XML documents by structure.

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

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

Parsons, L., Haque, E., and Liu, H. (2004).
Evaluating subspace clustering algorithms.
In Workshop on Clustering High Dimensional Data and its Applications, SIAM Int. Conf. on Data Mining, pages 48–56.

Quinlan, R. (2004).
Data mining tools see5 and c5.0.

Termier, A., Rousset, M.-C., and Sebag, M. (2002).
Treefinder : a first step towards xml data mining.
In IEEE International Conference on Data Mining (ICDM02), pages 450–457.

Ye, L. and Spetsakis, M. (Oct. 2003).
Clustering on unobserved data using mixture of gaussians.
Technical report, York University, Toronto, Canada.

Zaki, M. J. and Aggarwal, C. C. (2003).
Xrules : An effective structural classifier for xml data.
In SIGKDD 03, Washington, DC.