Learning Lexicographic Preference Trees from Positive Examples

Hélène Fargier **Pierre-François Gimenez** Jérôme Mengin

IRIT-CNRS
University of Toulouse

AAAI'18 Technical Track — February 2-7 2018, New Orleans

Context

Imagine a user wants to book a dinner online. A menu is a tuple of values of three different attributes:

Main course (M)	meat or fish
Wine (W)	red or white
Cheese (C)	goat or Camembert

During her choice, we would like to make a recommendation
We have access to a set of previously menus sold by the website

Example

Sales history
meat – red – goat
meat – red – goat
meat – red – Cam.
meat – red – Cam.
meat – red – Cam.
meat – white – goat
meat – white – Cam.
fish – white – goat
fish – white – Cam.

Example

Sales history
meat – red – goat
meat – red – goat
meat – red – Cam.
meat – red – Cam.
meat – red – Cam.
meat – white – goat
meat – white – Cam.
fish – white – goat
fish – white – Cam.

- Meat more common than fish: meat is probably preferred to fish

$$meat > fish$$

Example

Sales history
meat – red – goat
meat – red – goat
meat – red – Cam.
meat – red – Cam.
meat – red – Cam.
meat – white – goat
meat – white – Cam.
fish – white – goat
fish – white – Cam.

- Meat more common than fish: meat is probably preferred to fish

$$meat > fish$$

- For meat dinner, red wine seems preferred to white wine

$$meat : red > white$$

Example

Sales history
meat – red – goat
meat – red – goat
meat – red – Cam.
meat – red – Cam.
meat – red – Cam.
meat – white – goat
meat – white – Cam.
fish – white – goat
fish – white – Cam.

- Meat more common than fish: meat is probably preferred to fish

$$meat > fish$$

- For meat dinner, red wine seems preferred to white wine

$$meat : red > white$$

We can deduce information about user preferences !

Probabilistic model

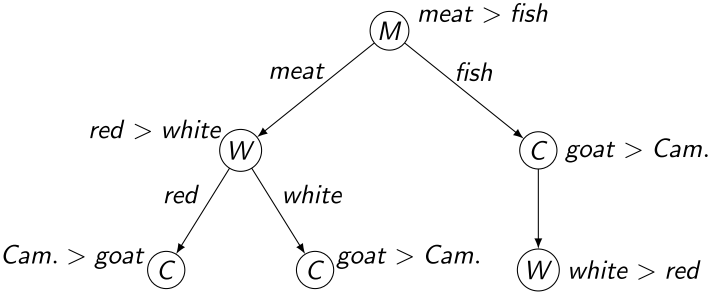
We don't always choose our most preferred outcome
(e.g. because of a desire of variety)

Ground idea

- The more preferred an outcome is, the more often it is chosen
- **Probability distribution of selection** p **decreasing w.r.t. the preference relation** $\succ$**:** $p(o) > p(o')$ **iff** $o \succ o'$

Idea not tied to any specific language
In the following: represented by lexicographic preference trees

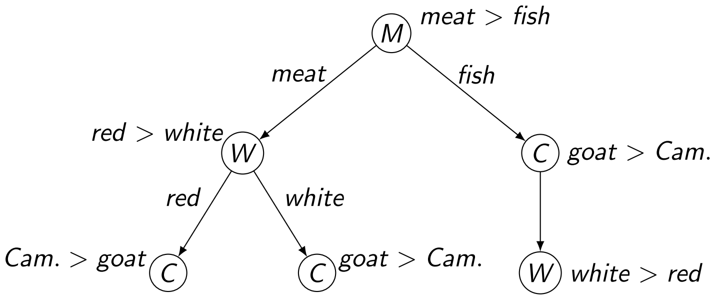
Lexicographic Preference Trees (LP-trees) [BCL+09]



LP-tree definition [BCL+09]

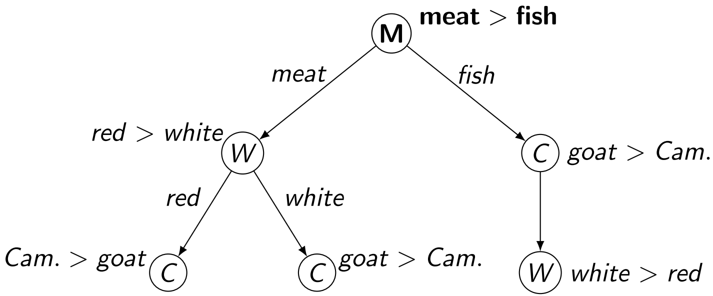
- Tree of attributes ordered by importance (root: most important)
- Edges can be labelled by a value or not
- Preferences rules associated to each node

LP-trees semantics



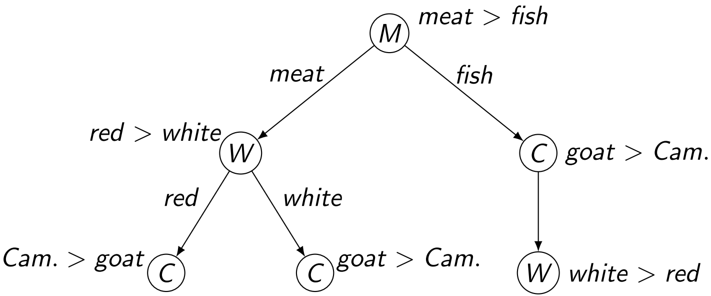
We would like to compare:
meat – red – goat and *fish – white – goat*

LP-trees semantics



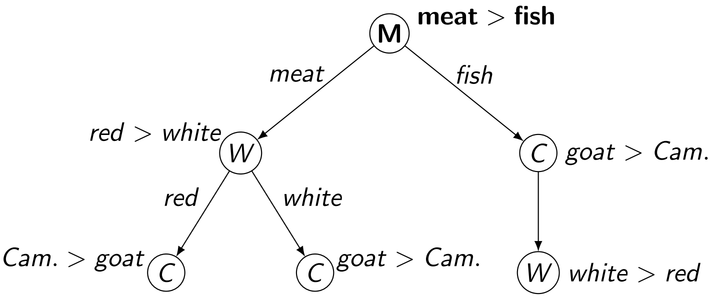
We would like to compare:
meat – *red* – *goat* $\succ$ *fish* – *white* – *goat*
Any menu with meat is preferred to any menu with fish

LP-trees semantics



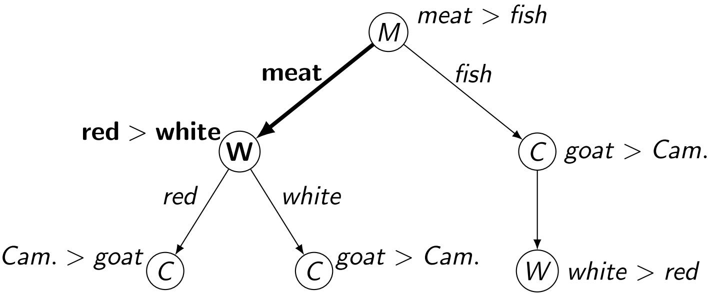
We would like to compare:
meat – white – goat and *meat – white – Cam.*

LP-trees semantics



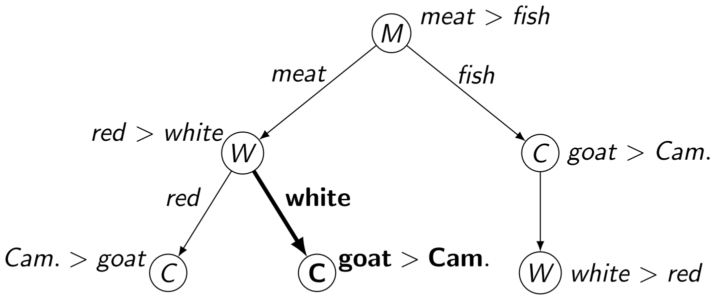
We would like to compare:
meat – *white* – *goat* and *meat* – *white* – *Cam.*
Root node can't decide the comparison

LP-trees semantics



We would like to compare:
meat – white – goat and *meat – white – Cam.*
Among meat menus, any menu with red wine is preferred to any menu with white wine

LP-trees semantics



We would like to compare:
$meat - white - goat \succ meat - white - Cam.$

Our contribution

Learning algorithms assume pairwise comparisons [BCL+09, BCL+10, BH12, LT15, BHKG17]

Here, no pairwise comparisons but sales histories

Our contribution

An algorithm to learn a LP-tree from sales histories

How to learn a LP-tree from a sales history ?

Sales history
meat – red – goat
meat – red – goat
meat – red – Cam.
meat – red – Cam.
meat – red – Cam.
meat – white – goat
meat – white – Cam.
fish – red – goat
fish – white – goat

How to learn a LP-tree from a sales history ?

Sales history
meat – red – goat
meat – red – goat
meat – red – Cam.
meat – red – Cam.
meat – red – Cam.
meat – white – goat
meat – white – Cam.
fish – red – goat
fish – white – goat

M	7 meat	2 fish
W	6 red	3 white
C	5 goat	4 Cam.

1. **Most important attribute**
= most unbalanced attribute

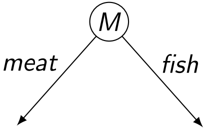
How to learn a LP-tree from a sales history ?

Sales history
meat – red – goat
meat – red – goat
meat – red – Cam.
meat – red – Cam.
meat – red – Cam.
meat – white – goat
meat – white – Cam.
fish – red – goat
fish – white – goat

M	7 meat	2 fish
W	6 red	3 white
C	5 goat	4 Cam.

1. **Most important attribute**

= most unbalanced attribute

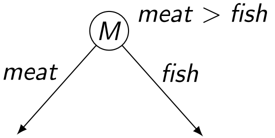


How to learn a LP-tree from a sales history ?

Sales history
meat – red – goat
meat – red – goat
meat – red – Cam.
meat – red – Cam.
meat – red – Cam.
meat – white – goat
meat – white – Cam.
fish – red – goat
fish – white – goat

M	7 meat	2 fish
W	6 red	3 white
C	5 goat	4 Cam.

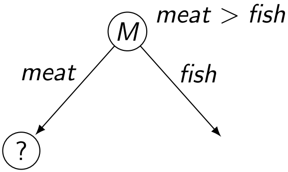
2. Preference relation



How to learn a LP-tree from a sales history ?

Sales history
meat – red – goat
meat – red – goat
meat – red – Cam.
meat – red – Cam.
meat – red – Cam.
meat – white – goat
meat – white – Cam.
fish – red – goat
fish – white – goat

3. Most important attribute for *meat* menus

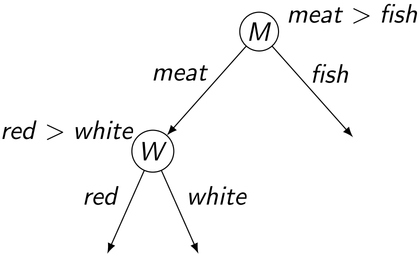


How to learn a LP-tree from a sales history ?

Sales history
meat – red – goat
meat – red – goat
meat – red – Cam.
meat – red – Cam.
meat – red – Cam.
meat – white – goat
meat – white – Cam.
fish – red – goat
fish – white – goat

W	5 red	2 white
C	3 goat	4 Cam.

3. Most important attribute for *meat* menus

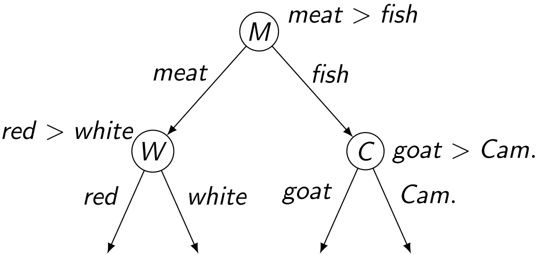


How to learn a LP-tree from a sales history ?

Sales history
meat – red – goat
meat – red – goat
meat – red – Cam.
meat – red – Cam.
meat – red – Cam.
meat – white – goat
meat – white – Cam.
fish – red – goat
fish – white – goat

W	1 red	1 white
C	2 goat	0 Cam.

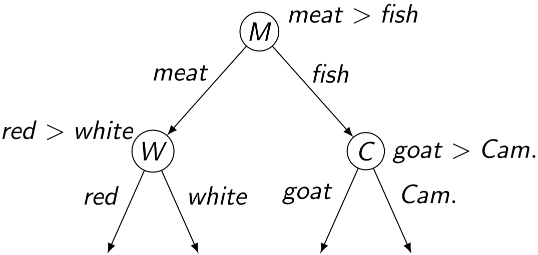
4. Most important attribute for *fish* menus



How to learn a LP-tree from a sales history ?

5. No exemple in the branch *fish* – *Cam.* !

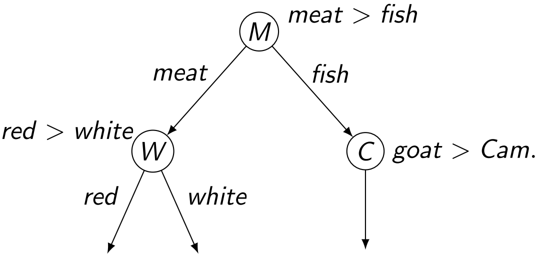
Sales history
meat – *red* – *goat*
meat – *red* – *goat*
meat – *red* – *Cam.*
meat – *red* – *Cam.*
meat – *red* – *Cam.*
meat – *white* – *goat*
meat – *white* – *Cam.*
fish – *red* – *goat*
fish – *white* – *goat*



How to learn a LP-tree from a sales history ?

Sales history
meat – red – goat
meat – red – goat
meat – red – Cam.
meat – red – Cam.
meat – red – Cam.
meat – white – goat
meat – white – Cam.
fish – red – goat
fish – white – goat

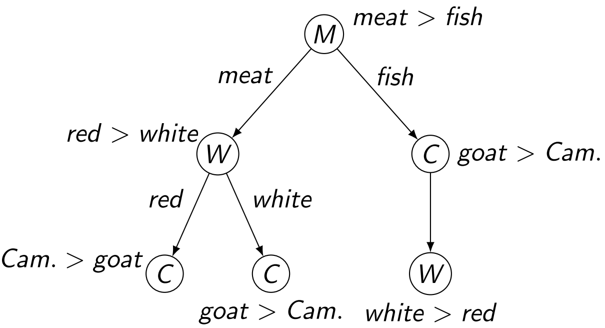
6. Solution: one unlabelled, unconditioned edge



How to learn a LP-tree from a sales history ?

7. And so on

Sales history
meat – red – goat
meat – red – goat
meat – red – Cam.
meat – red – Cam.
meat – red – Cam.
meat – white – goat
meat – white – Cam.
fish – red – goat
fish – white – goat



Algorithm

Algorithm 1: LP-tree learning algorithm

Input: $\mathcal{X}$, a set of outcomes $\mathcal{H}$ over $\mathcal{X}$
Output: $\mathcal{L}$ the learnt k-LP-tree
Algorithm `LearnLPTree`$(\mathcal{X}, \mathcal{H})$

- $\mathcal{L} \leftarrow$ unlabelled root node
- **while** $\mathcal{L}$ *contains some unlabelled node* N **do**
  - $(\mathbf{X}, \mathit{table}) \leftarrow$ ChooseAttributes(N)
  - label N with attributes $\mathbf{X}$ and CPT *table*
  - $L \leftarrow$ GenerateLabels$(N, \mathbf{X})$
  - **for each** $l \in L$ **do** add new unlabelled node to $\mathcal{L}$, attached to N with edge labelled with l
- **return** $\mathcal{L}$

Properties

Time complexity

For n attributes and a sales history $\mathcal{H}$, the time complexity is:

$$O(n^2|\mathcal{H}|^2)$$

Property 1

This algorithm converges to the target LP-tree as the sample size tends to infinity

Property 2

This algorithm finds the most probable linear LP-tree

Experiments on generated data: protocol

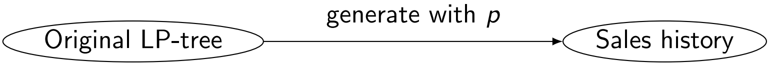
Experimental protocol

- LP-trees are randomly generated
- Sales histories are drawn from a geometric distribution p
- LP-trees are learnt from the sales histories
- Learnt LP-trees are compared with hidden LP-trees

Experiments on generated data: protocol

Experimental protocol

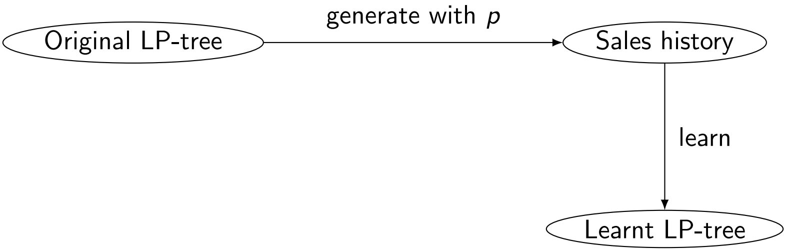
- LP-trees are randomly generated
- Sales histories are drawn from a geometric distribution p
- LP-trees are learnt from the sales histories
- Learnt LP-trees are compared with hidden LP-trees



Experiments on generated data: protocol

Experimental protocol

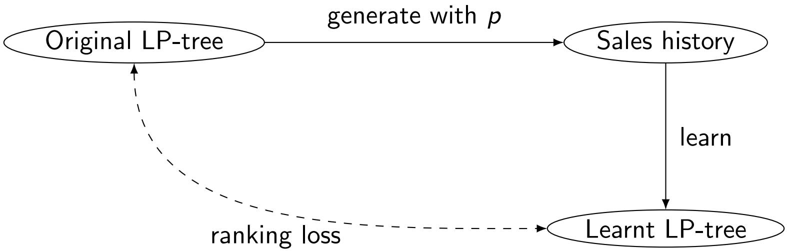
- LP-trees are randomly generated
- Sales histories are drawn from a geometric distribution p
- LP-trees are learnt from the sales histories
- Learnt LP-trees are compared with hidden LP-trees



Experiments on generated data: protocol

Experimental protocol

- LP-trees are randomly generated
- Sales histories are drawn from a geometric distribution p
- LP-trees are learnt from the sales histories
- Learnt LP-trees are compared with hidden LP-trees



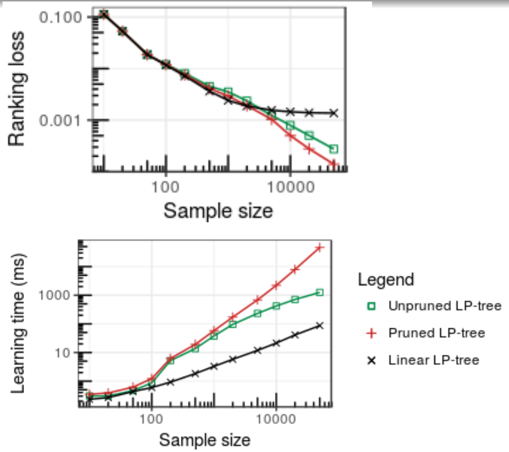
Experimental evaluation

Items in the sales history are probably ranked high in user preferences
A good LP-tree should rank high the items of the sales history

Evaluation of the LP-tree learnt

- Induction principle: minimize mean rank of items in the sales history
- **Ranking loss** = normalized difference of mean rank of items in the learnt LP-tree and the target LP-tree

Experiments on generated data: results



Results

- Quick convergence w.r.t. sample size: ranking loss seems inversely proportional to the sample size

Recommendation in car interactive configuration: dataset

Dataset

- Genuine sales history from Renault (car manufacturer)
- 48 attributes (mostly binary)
- 27088 items in sales history

From this sales history, we learn a LP-tree

Recommendation in car interactive configuration: protocol

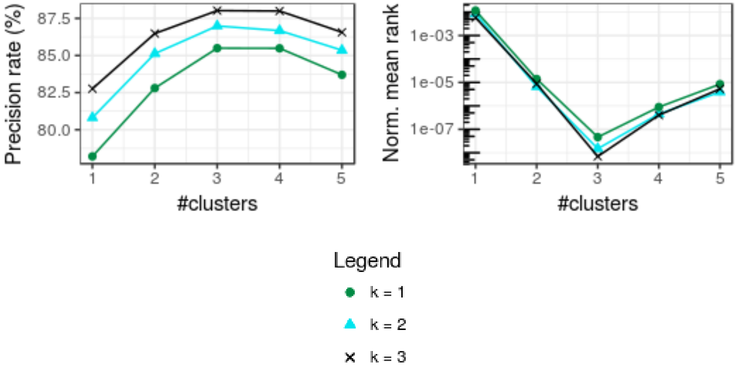
Interactive configuration

1. The user selects freely an attribute
2. The recommender system recommends a value
3. The user accepts the recommended value or chooses another one
4. Repeat until all attributes have a value

Protocol

- For each car in the test set, we simulate a configuration session
- Recommendation precision: ratio of recommendations that would have been accepted

Recommendation in car interactive configuration: results



Results

- Mean rank correlated with measured precision
- High precision (87% accepted recommendations)

Conclusion

Contributions

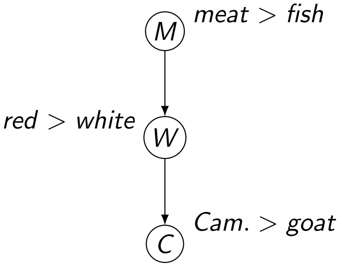
- Ground idea: preferred outcomes are more probably picked
- Framework and algorithm to learn LP-trees from **positives examples**
- Effective learning of randomly generated LP-trees
- Good recommendation precision on a real-world application

Perspectives

- Sample complexity in PAC settings
- Extension to other preference languages (e.g. CP-nets)

Linear LP-trees

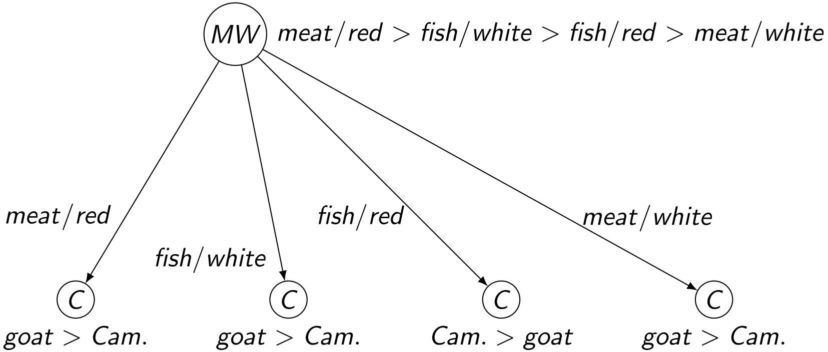
A linear LP-tree is a LP-tree with only unlabelled edges (i.e. a linear tree)
It is the classical "lexicographic order".



k-LP-trees

k-LP-trees may have at most k attributes per node (classical LP-trees are 1-LP-trees)

It can represent preference order where classical LP-trees can't

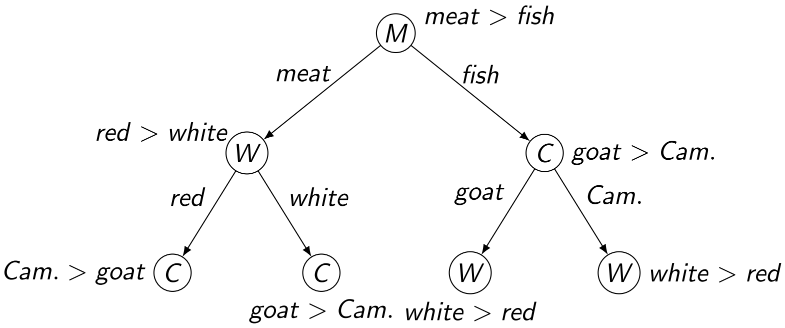


LP-tree pruning

The LP-tree learnt may overfit (learn by heart) the data, which decrease its generalization power

The pruning reduces overfitting by simplifying the LP-tree

Before pruning

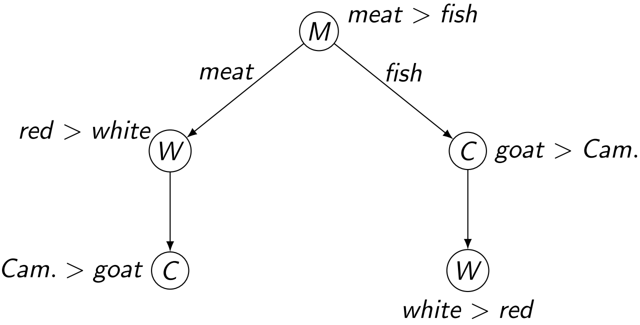


LP-tree pruning

The LP-tree learnt may overfit (learn by heart) the data, which decrease its generalization power

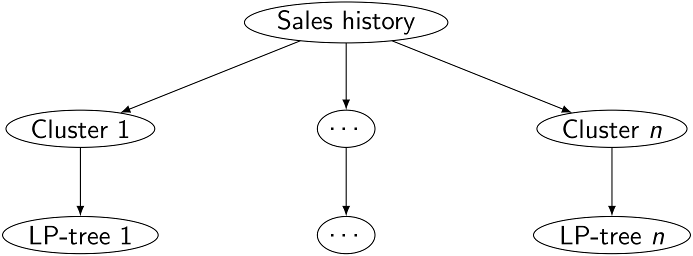
The pruning reduces overfitting by simplifying the LP-tree

After pruning



Clustering

We divide the sales history into homogeneous clusters
We learn a LP-tree for each cluster



Then, to make a recommendation given a partial assignment $\mathbf{u}$, we use the LP-tree whose cluster centre is closest (Hamming distance) to $\mathbf{u}$

Bibliographie I

- Richard Booth, Yann Chevaleyre, Jérôme Lang, Jérôme Mengin, and Chattrakul Sombattheera.
  Learning various classes of models of lexicographic orderings.
  Preference learning, page 1, 2009.

- Richard Booth, Yann Chevaleyre, Jérôme Lang, Jérôme Mengin, and Chattrakul Sombattheera.
  Learning conditionally lexicographic preference relations.
  In *Proceedings of ECAI'10*, pages 269–274, 2010.

- Michael Bräuning and Eyke Hüllermeyer.
  Learning conditional lexicographic preference trees.
  In Johannes Fürnkranz and Eyke Hüllermeyer, editors, *Proceedings of ECAI'12 Workshop*, 2012.

Bibliographie II

Michael Bräuning, Eyke Hüllermeier, Tobias Keller, and Martin Glaum.
Lexicographic preferences for predictive modeling of human decision making: A new machine learning method with an application in accounting.
European Journal of Operational Research, 258(1):295–306, 2017.

Xudong Liu and Miroslaw Truszczynski.
Learning partial lexicographic preference trees over combinatorial domains.
In *Proceedings of AAAI'15*, volume 15, pages 1539–1545, 2015.