Learning various classes of models of lexicographic orderings

Richard Booth[1], Yann Chevaleyre[2], Jérôme Lang[2],
Jérôme Mengin[3*], and Chattrakul Sombattheera[1]

[1] Mahasarakham University, Thailand
[2] LAMSADE, Université Paris-Dauphine, France
[3] IRIT, Université de Toulouse, France

Abstract. We consider the problem of learning a user's ordinal preferences on multiattribute domains, assuming that the user's preferences may be modelled as a kind of *lexicographic* ordering. We introduce a general graphical representation called *LP-structures* which captures various natural classes of such ordering in which both the order of *importance* between attributes and the *local preferences* over each attribute may or may not be conditional on the values of other attributes. For each class we determine the Vapnik-Chernovenkis dimension, the communication complexity of learning preferences, and the complexity of identifying a model in the class consistent with some given user-provided examples.

CNRS - INPT - UPS - UT1 - UTM

Rapport de Recherche IRIT/RR–2009-21–FR June 23, 2009

* Corresponding author

1 Introduction

In many applications, especially electronic commerce, it is important to be able to learn the preferences of a user on a set of alternatives that has a combinatorial (or multiattribute) structure: each alternative is a tuple of values for each of a given number of variables (or attributes). Whereas learning *numerical* preferences (*i.e.*, utility functions) on multiattribute domains has been considered in various places, learning *ordinal* preferences (*i.e.*, order relations) on multiattribute domains has been given less attention. Two streams of work are worth mentioning.

First, a series of very recent works focus on the learning of preference relations enjoying some preferential independencies conditions. Passive learning of separable preferences is considered by Lang & Mengin (2009), whereas passive (resp. active) learning of acyclic CP-nets is considered by Dimopoulos *et al.* (2009) (resp. Koriche & Zanuttini, 2009).

The second stream of work, on which we focus in this paper, is the class of *lexicographic* preferences, considered in Schmitt & Martignon (2006); Dombi *et al.* (2007); Yaman *et al.* (2008). These works only consider very simple classes of lexicographic preferences, in which both the importance order of attributes and the local preference relations on the attributes are unconditional. These very simple lexicographic preference models exclude the possibility to represent some more complex, yet natural, relations between objects. Suppose for instance that you want to buy a computer at a simple e-shop. Assuming your cash is not unlimited, the website first asks you to enter the maximum price you can afford to pay (for simplicity, we suppose here that this is not conditioned by the computer that you may buy). The objective of the website is to find the best (according to your preferences) computer you can afford. Suppose first that you always prefer laptops to desktop computers: the distinction between laptop and desktop makes the most important attribute to order computers according to your taste. Now, there are two other important criteria: the color of the computer, and whether it has a simple DVD-reader or a powerful DVD-writer. The color may be more important than the type of optical drive in the case of a laptop, because you would not want to be seen at a meeting with the usual bland, black laptop; in fact, you always prefer a flashy yellow laptop to a black one – whereas it is the opposite with desktops, because working long hours in front of a yellow desktop may be a strain for your eyes. Interestingly, this examples indicates that both the importance of the attributes and the local preference on the values of some attributes may be conditioned by the values of some other attributes: here, the relative importance of the color and the type of optical drive depends on the type of computer; and the preferred color depends on the type of computer as well.

In this paper we go further and consider various classes of lexicographic preference models, where the importance relation between attributes and/or the local preference on an attribute may depend on the values of some more important attributes. In Section 2 we give a general model for lexicographic preference relations, and define six classes of lexicographic preference relations, only two of which have already been considered from a learning perspective. Then each of the following sections focuses on a specific kind of learning problem: in Section 3 we address the sample complexity of learning lexicographic preferences, in Section 4 we consider preference elicitation, *a.k.a.* active learning, and in Section 5 we consider passive learning, and more specifically model identification and approximation.

2 Lexicographic preference relations: a general model

2.1 Lexicographic preferences structures

We consider a set $\mathcal{A}$ of n attributes, also called variables. Each attribute $X \in \mathcal{A}$ has an associated finite domain $\underline{X}$. We assume the domains of the various attributes are disjoint.

An attribute X is binary if its domain contains exactly two values, which by convention are denoted by x and $\overline{x}$. If $U \subseteq \mathcal{A}$ is a subset of the attributes, then $\underline{U}$ is the cartesian product of the domains of the attributes in U. Attributes, as well as sets of attribute, are denoted by upper-case Roman letters (X, X_i, A etc.) and attribute values by lower-case Roman letters. An outcome is an element of $\underline{\mathcal{A}}$; we will denote outcomes using greek lower case Greek letters (α, β, etc.).

Given a (partial) assignment $u \in \underline{U}$ for some $U \subseteq \mathcal{A}$, and $V \subseteq \mathcal{A}$, we denote by $u(V)$ the assignment made by u to the attributes in $U \cap V$.

Lexicographic comparison is a general way of ordering any pair of outcomes $\{\alpha, \beta\}$ by looking at the attributes in sequence, until one attribute X is reached such that α and β have different values of X: $\alpha(X) \neq \beta(X)$; the two outcomes are then ordered according to the *local preference* relation over the values of this attribute. Such a comparison uses two types of relation: a relation of *importance* between attributes, and *local preference* relations over the domain of each attribute.

Both the importance between attributes and the local preferences may be conditional. In the introductory example, if two computers share the value l (for laptop) for the attribute T (type), then C (color) is more important than D (the type of optical Drive), and y (yellow) is preferred to b (black); whereas when comparing computers of type d (desktops), D is more important than C, and b is preferred to y. Note that the condition on the type of computer assumes here that the two objects have the same value for this attribute. In this paper, we will only consider this simple type of conditions, which implies that the attributes that appear in the condition (T on the example) must be more important than the attribute over which a local preference is expressed (C) or the attributes, the importance of which is compared (C and D). [4]

Importance between attributes is captured by *Attribute Importance Trees:*

Definition 1. *An* Attribute Importance Tree *(or AI-tree for short) over set of attributes $\mathcal{A}$ is a tree whose nodes are labelled with attributes, such that no attribute appears twice on the same branch, and such that the edges between a non-leaf node n, labelled with attribute X, and its children are labelled with disjoint sets of values of X. An AI-tree is* complete *if every attribute appears in every one of its branches.*

For the sake of clarity, if one edge is labelled with the entire domain of an attribute X, we can omit this label (the labels on the edges are there to choose how to descend the tree according to the values of the attribute, and an edge labelled with the full domain of an attribute means there is no choice – note that there is no other "sibling" edge in this case since labels of different edges must be disjoint). Also, if an edge is labelled with a singleton $\{x\}$, we will often refer to the label by the value x itself.

We will denote by $\mathsf{Anc}(n)$ the set of ancestors of node n, that is the nodes on the path from the root to the parent of n. We will often identify $\mathsf{Anc}(n)$ with the set of attributes that label the ancestor nodes of n. We will denote by $\underline{n}$ the cartesian product of the labels of the edges on the path from the root to n.

Let us now turn to the representation of the local preferences on each attribute. When we want to compare two outcomes α and β using a lexicographic ordering, we go down the tree until we reach a node labelled with an attribute X that has different values in α and β: at this stage, we must be able to choose between the two outcomes according to some ordering over X.

[4] Exploring the possibility to have the local preference on an attribute domain depend on the value of a less important attribute is an interesting research direction, but it leads to many problems, starting from the fact that it may fail to be fully defined: take $\alpha = x_1x_2$, $\beta = \overline{x_1x_2}$, X_2 being more important than X_1, and assume we have this local preference relation for X_2: x_2 is preferred to $\overline{x_2}$ if $X_1 = x_1$ and $\overline{x_2}$ is preferred to x_2 if $X_1 = \overline{x_1}$. The most important attribute on which α and β differ is X_2, however the values of X_1 in α and β differ, therefore the local preference rules do not allow to order α and β. In other cases, preference cycles may appear.

Definition 2. *A* Local Preference Rule *(for attribute X, over set of attributes $\mathcal{A}$) is an expression $X, u :>$ where $u \in \underline{U}$ for some $U \subseteq \mathcal{A}$, $X \in \mathcal{A} - U$, and $>$ is a total linear order over $\underline{X}$. A* Local Preference Table *(over set of attributes $\mathcal{A}$) is a set of local preference rules.*

So far, importance trees and local preference tables have been defined independently. Now, as we said above, we require that the local preference relation for an attribute depends only on the values of more important attributes. For this we need the following definition:

Definition 3. *Let T be an AI-tree, n a node of T, and P a local preference table. A rule $X, v :>$ of P is said to be* applicable at node n given assignment $u \in \underline{n}$ *if (a) n is labelled by X and (b) $v \subseteq u$. P is* unambiguous w.r.t. T *(resp.* complete w.r.t. T*) if for any node n of T and any $u \in \underline{n}$, there is at most one (resp. exactly one) rule applicable at n given u.*

Definition 4. *A* Lexicographic Preference Structure *(or LP-structure) is a pair (T, P) where T is an attribute importance tree and P an unambiguous local preference table w.r.t. T. If furthermore T is complete and P is complete w.r.t. T, then (T, P) is a* Complete Lexicographic Preference Structure*.*

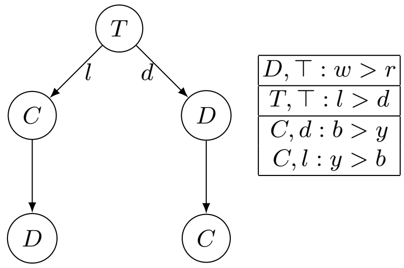

Fig. 1. Graphical representation of lexicographic orderings for Example 1

Example 1. Consider three attributes C(olor) with two values y(ellow) and b(lack), D(vd device) with two values w(riter) and r(ead-only), and T(ype) with values l(aptop) and d(deskop). The LP-structure σ depicted on Fig. 1 is a model for preferences about computers, where the type of computer is the most important criteria, with laptops always preferred to desktops, and where the second criterium is color in the case of laptops, with yellow laptops preferred to black ones, whereas the second criterium is the type of optical drive in the case of desktops. In any case, a writer is always preferred to a read-only drive. The color is third criteria for desktops, with black preferred to yellow in this case.

The semantics of LP-structures is defined by the associated orderings over outcomes:

Definition 5. *LP-structure $\sigma = (T, P)$ defines a partial strict order $>_\sigma$ over the set of outcomes as follows: given any pair of outcomes $\{\alpha, \beta\}$, go down the tree, starting at the root, following edges that correspond to assignments made in α and β, until the first node n is reached that is labelled with attribute X such that $\alpha(X) \neq \beta(X)$; we say that n* decides *$\{\alpha, \beta\}$. If there is a rule $X, v :>$ in P that is applicable at n given $u = \alpha(\mathsf{Anc}(n)) = \beta(\mathsf{Anc}(n))$, then $\alpha >_\sigma \beta$ if and only if $\alpha(X) > \beta(X)$. If there is no rule that is applicable at n given u, or if no node that decides $\{\alpha, \beta\}$ is reached, α and β are σ-incomparable.*

Example 1 (continued). According to σ, the most preferred computers are yellow laptops with a DVD-writer, because $ywl >_\sigma \alpha$ for any other outcome $\alpha \neq ywl$; for any $x \in \underline{C}$ and any $z \in \underline{D}$ $xzl >_\sigma xzd$, that is, any laptop is preferred to any desktop computer. And $ywd >_\sigma brd$, that is, a yellow deskop with DVD-writer is preferred to a black one with DVD-reader because, although for desktops black is preferred to yellow, the type of optical reader is more important than the colour for desktop computers.

Proposition 1. *Given a LP-structure $\sigma = (T, P)$, the relation $>_\sigma$ is irreflexive and transitive. It is also modular, i.e., $\alpha >_\sigma \beta$ implies either $\alpha >_\sigma \gamma$ or $\gamma >_\sigma \beta$. Moreover, if σ is complete, then $>_\sigma$ is a linear order.*

Proof. Irreflexivity is trivial: there is no node that decides a pair $\{\alpha, \alpha\}$. For transitivity, let α, β and γ be three outcomes such that $\alpha >_\sigma \beta >_\sigma \gamma$. Let m be the node that decides $\{\alpha, \beta\}$, labelled with attribute X, and let n be the node that decides $\{\beta, \gamma\}$. Note that m and n are both on the branch that corresponds to the assignments made by β. Let $u = \alpha(\mathsf{Anc}(m)) = \beta(\mathsf{Anc}(m))$, and let $X, v :>$ be the rule in P that is applicable at m given u. If $m = n$, then $u = \gamma(\mathsf{Anc}(m))$, and $\alpha(X) > \beta(X) > \gamma(X)$, hence $\alpha >_\sigma \beta$. If m is above n, then $\alpha(X) > \beta(X) = \gamma(X)$, so m decides $\{\alpha, \gamma\}$, and $\alpha >_\sigma \gamma$. If m is below n, a similar proof holds.

For modularity suppose $\alpha >_\sigma \beta$ and let n be the node which decides $\{\alpha, \beta\}$. Suppose n is labelled by attribute X, and let $X, u :>$ be the only rule of P that applies at n given $u = \alpha(\mathsf{Anc}(m)) = \beta(\mathsf{Anc}(m))$: then $\alpha(X) > \beta(X)$. We consider four cases:

1. There is no node which decides $\{\alpha, \gamma\}$. Then $\beta(Y) = \alpha(Y) = \gamma(Y)$ for all $Y \in \mathsf{Anc}(n)$ and $\alpha(X) = \gamma(X) > \beta(X)$, hence n decides $\{\gamma, \beta\}$ and $\gamma >_\sigma \beta$.
2. The node which decides $\{\alpha, \gamma\}$ is below n. Then the same reasoning as Case 1 yields $\gamma >_\sigma \beta$.
3. Node n decides $\{\alpha, \gamma\}$. If $\gamma(X) = \beta(X)$ then $\alpha(X) > \beta(X)$ yields $\alpha(X) > \gamma(X)$ and so $\alpha >_\sigma \gamma$. If $\gamma(X) \neq \beta(X)$ then n also decides $\{\beta, \gamma\}$. In this case the facts that $\alpha > \beta$ and $>$ is a strict total order (hence itself modular) means we must have either $\alpha > \gamma$ or $\gamma > \beta$, i.e., $\alpha >_\sigma \gamma$ or $\gamma >_\sigma \beta$.
4. The node m which decides $\{\alpha, \gamma\}$ is above n. Let Z be the label of this deciding node. Then since $\alpha(Y) = \beta(Y)$ for all $Y \in \mathsf{Anc}(n)$, m also decides $\{\beta, \gamma\}$. Let $Z, v :>$ be the rule that is applicable at m with respect to $u = \alpha(\mathsf{Anc}(m)) = \beta(\mathsf{Anc}(m)) = \gamma(\mathsf{Anc}(m))$. Since $>$ is a strict total order we know either $\alpha(Z) > \gamma(Z)$ or $\gamma(Z) > \alpha(Z) = \beta(Z)$. The first case gives $\alpha >_\sigma \gamma$ and the latter gives $\gamma >_\sigma \beta$.

Finally, if the structure σ is complete, every pair of outcomes is decided at some node, thus $>_\sigma$ is a linear order.

The above proposition is saying $>_\sigma$ is a modular strict partial order. Every such order can be seen as the strict version of a total preorder. This means that even when $>_\sigma$ is not a linear order, it may still be viewed as "ranking" the different outcomes, with outcomes which are σ-incomparable given the same rank. (To be more precise, the relation $\geq_\sigma$ defined by $\alpha \geq_\sigma \beta$ iff [$\alpha >_\sigma \beta$ or α, β are σ-incomparable] is a total preorder.)

2.2 Classes of lexicographic preference structures

We now define interesting classes of LP-structures.

Classes of LP-structures with conditional preferences It should be clear that any LP-structure σ is equivalent to a LP-structure σ' where each edge corresponds to exactly one value of its parent node, and where each preference rule applies to exactly one node: σ' can be obtained from σ by multiplying the edges that correspond to more than one value; and by multiplying the preference rules that apply at more than one node. This structure σ' can be seen as a canonical representation of $>_\sigma$. This leads to the following defnition:

Definition 6. *A* CP&I *LP-structure, or structure with conditional local preferences and conditional attribute importance, is a structure in which each edge of the tree is labelled with a singleton value, and such that for each node n, that corresponds to exactly one partial assignment u, the preference table contains one rule of the form $X, u :>$ where X is the attribute that labels n.*

CP&I LP-structures are particular cases of Wilson's "Pre-Order Search Trees" (or POST) (2006): in POSTs, the preference relation at every node can be a non strict relation.

Example 1 (continued). A CP&I structure equivalent to the LP-structure depicted on Fig. 1 for Example 1 is depicted on Fig. 2. Note that when we draw a *CP&I* LP-structure, since local preferences at a given node can only depend on attributes above that node, we can represent the local preference relation corresponding to a node inside the node itself.

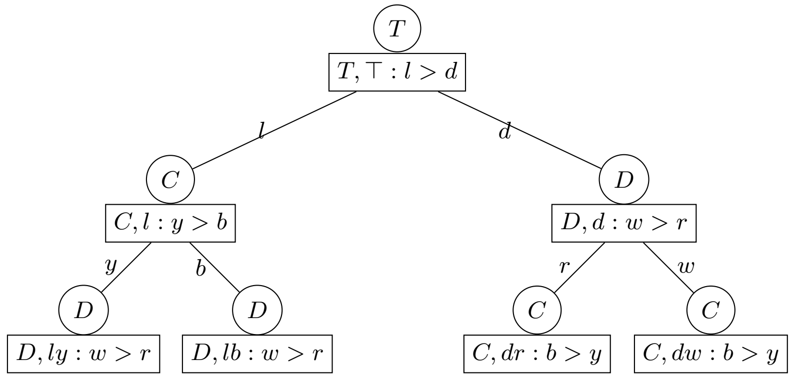

Fig. 2. A CP&I structure equivalent to that of Example 1

Another interesting class is that of structures with conditional preferences but *un*conditional attribute importance:

Definition 7. *A* CP-UI *LP-structure, or structure with conditional local preferences and* un*conditional attribute importance, is a structure in which the tree is linear, with each edge labelled with the full domain of the attribute at the parent node, and such that for each node n, for each partial assignment $u \in \underline{n}$, the preference table contains one rule of the form $X, u :>$ where X is the attribute that labels n.*

Classes of LP-structures with unconditional preferences We now turn to lexicographic preferences with unconditional preferences, like the ones studied by e.g. Schmitt & Martignon (2006); Dombi *et al.* (2007); Yaman *et al.* (2008):

Definition 8. *UP&I LP-structures, or structures with* un*conditional local preferences and* un*conditional attribute importance, are structures whose attribute importance tree is linear, each edge being labelled with the full domain of the attribute at the parent node, and whose preference table contains one unconditional rule of the form $X, \top :>$ for each attribute X that appears in the tree. UP-CI LP-structures, or structures with* un*conditional local preferences and conditional attribute importance, are structures in which each edge of the tree is labelled with a singleton value, so that each node corresponds to exactly one partial assignment, but such that for each attribute that appears in the tree, the local preference table contains only one unconditional rule of the form $X, \top :>$.*

We can also define classes of LP-structures with unconditional, **fixed** preferences:

Definition 9. *Given a non ambiguous set P of preferences rules, FP-UI(P) is the class of UP&I structures that have P for preference table. Similarly, FP-CI(P) is the class of UP-CI structures that have P for preference table.*

3 Sample complexity of some classes of LP-structures

3.1 The learning setting

Our aim in this paper is to study how we can learn a LP-structure that fits well some examples of comparison. We assume a set $\mathcal{E}$ of examples, that is, of pairs of outcomes over $\mathcal{A}$: we would like to find a LP-structure that is "consistent" with the examples in the following sense:

Definition 10. *LP-structure σ is said to be* consistent *with example $(\alpha, \beta) \in \underline{\mathcal{A}}^2$ if $\alpha >_\sigma \beta$; σ is consistent with set of examples $\mathcal{E}$ if it consistent with every example of $\mathcal{E}$.*

The problem of learning a structure that orders "well" the examples can be seen as a problem of classification: given σ we can define another binary relation $\leq_\sigma$ over $\underline{\mathcal{A}}^2$ as follows:

$$\alpha \leq_\sigma \beta \text{ if and only if } \beta >_\sigma \alpha \text{ or } (\alpha \not>_\sigma \beta \text{ and } \beta \not>_\sigma \alpha).$$

Because $>_\sigma$ is modular, $\leq_\sigma$ defined in this way is a total preorder over $\underline{\mathcal{A}}$ (i.e. the relation is reflexive, transitive, and for every $\alpha, \beta \in \underline{\mathcal{A}}$, at least one of $\alpha \leq_\sigma \beta$ or $\beta \leq_\sigma \alpha$ holds), and $\{\leq_\sigma, >_\sigma\}$ is a partition of $\underline{\mathcal{A}}^2$. In particular, we can define the Vapnik-Chernovenkis dimension of a class of LP-structures as the size of the biggest set of pairs (α, β) that can be "classified" correctly by some LP-structure in the class, whatever the labels ($>$ or $\leq$) associated with each pair. In general, the higher this dimension, the more examples will be needed to correctly identify a LP-structure.

In the next few sections we study the VC dimension of the classes of LP-structures defined in Section 2.2.

3.2 Classes of LP-structures with conditional preferences

Proposition 2. *The VC dimension of any class of transitive relations over a set of binary attributes is strictly less than 2^n.*

Proof. Any graph with N vertices and N edges contains at least one cycle. Consider a set of $N = 2^n$ pairs of outcomes over $\mathcal{A}$: since $\underline{\mathcal{A}}$ contains $N = 2^n$ outcomes, this set of pairs contains at least one cycle, that is, there is a sequence of at least two distincts pairs $\{\alpha_1, \alpha_2\}, \{\alpha_2, \alpha_3\}, \ldots, \{\alpha_{k-1}, \alpha_k\}$ with $\alpha_k = \alpha_1$; but then, no transitive relation $>$ over $\underline{\mathcal{A}}$ can be such that $\alpha_1 > \alpha_2 > \ldots > \alpha_{k-1} > \alpha_k$ and $\alpha_k > \alpha_1$ because $>$ is transitive. Thus it is not possible to shatter 2^n pairs of outcomes.

Proposition 3. *The VC dimension of both classes of CP&I LP-structures and of CP-UI structures over n* binary *attributes, is equal to $2^n - 1$.*

Lemma 1. *Let T be a AI tree over binary attributes; let $K = \sum_{n \text{ node of } T} |\underline{n}|$ be the sum of the cardinalities of the "domains" of the nodes of T: there is a set of K examples that is shattered by the set of LP-structures with conditional preferences and whose AI-tree is T.*

Proof. For every node n of T labelled with attribute X, for each $u \in \underline{n}$, define an unlabelled example $\{\alpha_u, \beta_u\}$ such that $\alpha_u = uxv$ and $\beta_u = u\overline{x}w$, for some partial assignments v and w. Given any labelling of this set of examples let P be the preference table that contains, for each $u \in \underline{n}$ for each node n, either $X, u : x > \overline{x}$ or $X, u : \overline{x} > x$, so that it is consistent with the labelling for $\{\alpha_u, \beta_u\}$: then (T, P) is consistent with the set of the labelled examples.

Proof of the proposition. It is clear that it is possible to build a AI tree over n attributes with 2^k nodes at the $k-th$ level, for $0 \leq i \leq n-1$, with $|\underline{n}| = 1$ for every node: this is a tree for CP&I structures, it has $2^n - 1$ nodes, thus it can shatter $2^n - 1$ examples. Consider now a linear tree, with the node n_k at the k-th level containing all possible assignments to its ancestor attributes: $|n_k| = 2^k$, thus it can shatter $\sum_{k=0}^{n-1} 2^k$ examples, hence the VC dimension of CP-UI structures is at least $2^n - 1$. The upper bound follows from Prop. 2.

This result is rather negative, since it indicates that a huge number of examples would in general be necessary to have a good chance of closely approximating an unknown target relation. This important number of necessary examples also means that it would not be possible to learn in reasonable - that is, polynomial - time. However, learning CP&I LP-structures is not hopeless in practice: decision trees have a VC dimension of the same order of magnitude, yet learning them has had great success experimentally.

3.3 Classes of LP-structures with unconditional preferences

Schmitt & Martignon (2006) have shown that hhe VC dimension of UP&I structures over n *binary* attributes is exactly n. Since every UP&I structure is equivalent to a CP-UI one, we obtain a lower bound on the VC dimension of the class of CP-UI structures over binary attributes:

Proposition 4. *The VC dimension of UP&I structures over* n binary *attributes is at least* n.

4 Preference elicitation/active learning

We now turn to the *active learning* of preferences. The setting is as follows: there is some unknown target preference relation $>$, and a *learner* wants to learn a representation of it by means of a Lexicographic Preference structure. There is a *teacher*, a kind of oracle to which the learner can submit queries of the form $\{\alpha, \beta\}$ where α and β are two outcomes: the teacher will then reply wether $\alpha > \beta$ or $\beta > \alpha$ is the case. An important question in this setting is: how many queries does the learner need in order to completely identify the target relation $>$? More precisely, we want to find the communication complexity of preference elicitation, i.e., the worst-case number of requests to the teacher to ask so as to be able to elicit the preference relation completely, assuming the target can be represented by a model in a given class. The question has already been answered in Dombi *et al.* (2007) for the FP-UI case. Here we identify the communication complexity of eliciting lexicographic preferences structures in all 5 other cases, when all attributes are binary. (We restrict to the case of binary attributes for the sake of simplicity. The results for nonbinary attributes would be similar.) We know that a lower bound of the communication complexity is the log of the number of preference relations in the class. In fact, this lower bound is reached in all 6 cases:

Proposition 5. *The communication complexities of the six problems above are as follows,* when all attributes are binary.

	FP	*UP*	*CP*
UI	$\log(n!)$ *Dombi* et al. *(2007)*	$n + \log(n!)$	$2^n - 1 + \log(n!)$
CI	$g(n) = \sum_{k=0}^{n-1} 2^k \log(n-k)$	$n + g(n)$	$2^n - 1 + g(n)$

Proof. In all cases, the lower bound is obtained by counting the preference relations contained in the class involved. If C is a class of preference relations, and $\#(C)$ its cardinality, then the communication complexity is at least $\log \#C$.

Recall that a conditionally lexicographic preference relation is defined as a pair (T, P) where T is an unconditional preference tree and N a collection of unconditional preference tables compatible with N. Let $\succ_{T,P}$ be the UP&I preference relation induced from T and P.

In the four cases FP-UI, UP&I, FP-CI and UP-CI, T and P are independent, *i.e.*, any P is compatible with any T. Furthermore, it is clear that we have $\succ_{T,P}=\succ_{T',P'}$ if and only if $T = T'$ and $P = P'$. Therefore, the cardinality of one of these four classes is the product of the number of possible importance trees corresponding to the class and the number of possible collections of preference tables corresponding to the class. Clearly, there are $n!$ unconditional importance trees, only one collection of fixed preference tables, and 2^n collection of unconditional preference tables. We have now to count the conditional importance trees. Note first that every conditional importance tree can be equivalently rewritten into a *normalized* preference tree where every edge is labelled by a single value attribute. Then, at each node n of a normalized preference tree, if the depth of the node is k (the root having depth 0, etc.) then there are $n - k$ possibilities for labelling n with an attribute. Because the attributes are all binary, the normalized tree also is binary, and there are 2^k nodes of depth k, therefore the number of possible trees is $\prod_{k=0}^{n-1}(n-k+1)^{2^k}$. Let $g(n) = \log(\prod_{k=0}^{n-1}(n-k+1)^{2^k}) = \sum_{k=0}^{n-1} 2^k . \log(n-k)$. We have now the following four lower bounds for communication complexity:

- FP-UI: $\log(n!)$;
- UP&I: $\log(n!.2^n) = \log(n!) + n$;
- FP-CI: $g(n)$;
- UP-CI: $g(n) + n$.

Now, consider a normalized conditional importance tree and let us count the number of possible collection of preference tables that are compatible with it. At each node n, labelled with attribute X_i, we have 2 possibilities for the preference relation on D_i (either $x_i \succ \overline{x_i}$ or $\overline{x_i} \succ x_i$). Now, the tree contains $\sum_{k=0}^{n-1} 2^k = 2^n - 1$ nodes, therefore the number of possible collections of tables compatible with T is 2^{2^n-1}. The communication complexity of $CI - CLP$ is at least $g(n) + 2^n - 1$. Finally, let us consider UI-CLP. The fact that the importance tree is unconditional does not make any difference as to the number of preference tables that are compatible with it. Therefore, the ommunication complexity of $UI - CLP$ is at least $\log(n!) + 2^n - 1$.

Now, let us show that these lower bounds are reached (in all 6 cases). For FP-UI, this is already known from Dombi *et al.* (2007). For all other 5 cases, we have to design an elicitation protocol that guaranteed to identifiy the preference structure. Rather than explicitly designing the protocol, we start by establishing an important lemma, from which all results follow:

- for UP&I: we start by identifying the local preferences: for every attribute X_i we ask the user to compare $(\boldsymbol{x}_{-i}x_i)$ and $(\boldsymbol{x}_{-i}\overline{x_i})$ for an arbitrary $\boldsymbol{x}_{-i}$; therefore n queries anr sufficient for eliciting the whole local preference structure. Then we identify the importance tree. For this we start by identifying the most important attribute, for which $\log n$ queries are sufficient. To take an example, suppose we have four attributes, and that the local preferences (already identified) are c Then we ask the user to compare $\alpha = x_1\overline{x_2}x_3\overline{x_4}$ and $\beta = \overline{x_1}x_2\overline{x_3}x_4$. Note that in α (resp. β), X_1 and X_2 (resp. X_3 and X_4) take their preferred values. Therefore, if the user prefers α to β (resp. β to α), the most important attribute is X_1 or X_2 (resp. X_3 or X_4), and one more query will identify it completely. The rest is similar: at each step we use a dichotomic protocol, whose total communucation complexity for eliciting the importance structure is $\log n + \log(n-1) + ... + 1$, that is, $\log(n!)$. The lower bound for UP&I follows.

- for FP-CI and FP-UI: things are similar, except that the search for the next important attribute has to be done for each value of the current attribute. Therefore, we must run the dichotomic elicitation protocol for every non-terminal node of the importance tree, and for every node the protocol uses $\log(n-k)$ queries, where k is the depth of the node. Therefore the idtnficiation of the tree takes $g(n)$ queries and the two lower bounds for FP-CI and FP-UI follow.
- for CP-UI, the trees and the local preference shave to be elicited together, and the elicitation protocol is more complex; we give here only a sketch of its description, using an example. Suppose we have four attributes. We start by asking the user to compare $\alpha = x_1x_2x_3x_4$ and $\beta = \overline{x_1x_2x_3x_4}$. If she prefers α (resp. β) then we know that the most important attribute X_i is such that x_i is preferrred to $\overline{x_i}$ (resp. $\overline{x_i}$ is preferrred to x_i) and we can now use the above dichotomic protocol to identify the most important attribute, which takes $1 + \log n$ queries. Assume w.l.o.g. that this attribute is X_1. Then weconsider the two cases $X_1 = x_1$ and $X_1 = \overline{x_1}$ in sequence. For $X_1 = x_1$ we use the same protocol as above (except that the next most important attribute cannot be X_1), which takes $1 + \log(n-1)$ queries. Now, we consider $X_1 = \overline{x_1}$. Because the tree is unconditional, we wlready know the most important attribute, therefore we just need to elicit its preferred value, which takes 1 query. In all, for identifying the second most important attribute and its local preference relation given the possible values of the most important attribute, we need $2 + \log(n-1)$ queries. For the third most important attribute, similarly, we need $4 + \log(n-2)$ queries, and so on. Hence the upper bound $2^{n-1} + log(n!)$.
- for CI-CLP, this is similar as for UI-CLP, except that the next most important attribute has to be elicited for every possible values of the attributes in the branch considered.

5 Model identifiability

We now turn to the problem of identifying a model of a given class $\mathcal{C}$, given a set $\mathcal{E}$ of examples: each example is a pair (α, β), for which we know that $\alpha > \beta$ for some target preference relation $>$. The aim of the learner is to find some LP-structure σ in $\mathcal{C}$ such that $\alpha >_\sigma \beta$ for every $(\alpha, \beta) \in \mathcal{E}$.

Dombi *et al.* (2007) have shown that the corresponding decision problem for the class of binary LP-structures with unconditional importance and unconditional, fixed local preferences can be solved in polynomial time: given a set of examples $\mathcal{E}$ and a set P of unconditional local preferences for all attributes, is there a structure in $\mathrm{FP} - \mathrm{UI}(P)$ that is consistent with $\mathcal{E}$? In order to prove this, they exhibit a simple greedy algorithm. We will prove in this section that the result still holds for most of our classes of LP-structures, except one.

5.1 A greedy algorithm

In order to prove this, we will prove that the greedy Algorithm 1, when given a set of examples $\mathcal{E}$, returns a LP-structure that satisfies the examples if one exists. The algorithm recursively constructs the AI-tree from the root to the leaves. At a given currently unlabelled node n, step 2b considers the set $\mathcal{E}(n) = \{(\alpha, \beta) \in \mathcal{E} \mid \alpha(\mathsf{Anc}(n)) = \beta(\mathsf{Anc}(n)) \in \underline{n}\}$ of examples that correspond to the assignments made in the branch so far and that are still undecided: it looks for some attribute $X \notin \mathsf{Anc}(n)$ that can be used to order well examples in $\mathcal{E}(n)$ that can be ordered with X: there must be a set of local preferences rules of the form $X, w :>$ that is not ambiguous when put together with the current set of rules, and such that for every $(\alpha, \beta) \in \mathcal{E}(n)$, if $\alpha(X) \neq \beta(X)$ then there is a rule $X, w :>$ with $w \subseteq \alpha(U) = \beta(U)$ and $\alpha(X) > \beta(X)$. The attribute X can then be chosen for the label of n, and the set of rules added to P. Step 2f then considers the values of X that

Algorithm 1 GenerateLPStructure

Input: $\mathcal{A}$: set of attributes; $\mathcal{E}$: set of examples over $\mathcal{A}$;
P: set of local preference rules: initially empty,
or contains a set of unconditional preference rules for the **FLP** cases;
Output: LP-structure consistent with $\mathcal{E}$, that contains P, or FAILURE;

1. $T \leftarrow \{$unlabelled root node$\}$;
2. while T contains some unlabelled node:
   (a) choose unlabelled node n of T;
   (b) $(X, \mathit{newRules}) \leftarrow \mathsf{chooseAttribute}(\mathcal{E}(n), \mathsf{Anc}(n), P)$;
   (c) if $X = \mathsf{FAILURE}$ then STOP and return $\mathsf{FAILURE}$;
   (d) label n with X;
   (e) $P \leftarrow P \cup \mathit{newRules}$;
   (f) $L \leftarrow \mathsf{generateLabels}(\mathcal{E}(n), X)$; *(Create set of labels for edges below n)*
   (g) for each $l \in L$:
   add new unlabelled node to T, attached to n with edge labelled with l;
3. return (T, P).

correspond to still undecided examples, and prepare labels that will be used for the edges from n to its children. P is initially empty except in the case where the local preferences are known in advance, with only the order of importance to be learned. Note that this aproach cannot work in the case of conditional importance and unconditional preferences, as will be proved in Corollary 1.

Let us briefly describe the helper functions that appear in the algorithm:

$\mathsf{generateLabels}$ should return a set of disjoint subsets of the domain of the attribute at the current node; it takes as parameters a set of examples $\mathcal{E}(n)$, and the attribute X at the current node: we require that for each example $(\alpha, \beta) \in \mathcal{E}(n)$ that cannot be decided at n because $\alpha(X) = \beta(X)$, there is one label returned by $\mathsf{generateLabels}$ that contains $\alpha(X)$.

We will use two particular instances of the function $\mathsf{generateLabels}$:

$\mathsf{generateCondLabels}(\mathcal{E}, X) =$
$\{\{x\} \mid x \in \underline{X}$ and there is $(\alpha, \beta) \in \mathcal{E}$ such that $\alpha(X) = \beta(X) = x\}$:
in the case of conditional importance, each branch corresponds to one value of X.

$\mathsf{generateUncondLabel}(\mathcal{E}, X) = \{\underline{X}\}$: in the case of unconditional importance, one branch is created, except that if there is no $(\alpha, \beta) \in \mathcal{E}$ such that $\alpha(X) = \beta(X) = x$, then $\mathsf{generateUncondLabel}(\mathcal{E}, X) = \emptyset$.

$\mathsf{chooseAttribute}$ takes as parameters the set of examples $\mathcal{E}(n)$ that correspond to the node being treated, the set of attributes $\mathsf{Anc}(n)$ that already appear on the current branch, and the current set of preference rules P; it returns an attribute X not already on the branch to n and a set *newRules* of local preference rules over X: the attribute and the rules should be chosen so that they will decide well some examples of $\mathcal{E}(n)$. More precisely, we will require that $(X, \mathit{newRules})$ is *choosable* with respect to $\mathcal{E}(n), \mathsf{Anc}(n), P$ in the following sense:

Definition 11. *Given a set of examples $\mathcal{E}$ over attributes $\mathcal{A}$, a set of attributes $U \subseteq \mathcal{A}$ and a set of local preference rules P, $(X, \mathit{newRules})$ is* choosable with respect to $\mathcal{E}, U, P$ *if $X \in \mathcal{A} - U$, newRules is a set of local preference rules for X, and:*

- *$P \cup \mathit{newRules}$ is not ambiguous;*
- *for every $(\alpha, \beta) \in \mathcal{E}$, if $\alpha(X) \neq \beta(X)$ then there is a (unique) rule $X, v :>$ in $P \cup \mathit{newRules}$ such that $v \subseteq \alpha(U)$, and $\alpha(X) > \beta(X)$.*

Moreover, we will say that $(X, \mathit{newRules})$ is:

UP-choosable *if it is choosable and newRules if of the form* $\{X, \top :>\}$ *(it contains a single unconditional rule);*

CP-choosable *if it is choosable and newRules contains one rule* $X, u :>$ *for every* $u \in \underline{U}$ *such that there exists* $(\alpha, \beta) \in \mathcal{E}$ *with* $\alpha(U) = \beta(U) = u$.

5.2 Some examples of GenerateLPStructure

In these examples we assume three binary attributes A, B, C. Throughout this subsection we assume the algorithm checks the attributes for choosability in the order $A \to B \to C$. Furthermore we assume we are not in the FP case, i.e., the algorithm initialises with an empty local preference table $P = \emptyset$.

Example 2. Suppose $\mathcal{E}$ consists of the following five examples:

$$1.\ (abc, a\bar{b}c)\ \ 2.\ (\bar{a}\bar{b}c, \bar{a}bc)\ \ 3.\ (ab\bar{c}, a\bar{b}\bar{c})\ \ 4.\ (\bar{a}\bar{b}\bar{c}, \bar{a}b\bar{c})\ \ 5.\ (\bar{a}\bar{b}c, \bar{a}b\bar{c})$$

Let's try using the algorithm to construct a UP&I structure consistent with $\mathcal{E}$. At the root node n_0 of the AI-tree we first check if $(A, \textit{newRules})$ is UP-choosable w.r.t. $\mathcal{E}, \emptyset, \emptyset$. By the definition of UP-choosability, *newRules* must be of the form $\{A, \top :>\}$ for some total order $>$ of $\{a, \bar{a}\}$. Now since $\alpha(A) = \beta(A)$ for all $(\alpha, \beta) \in \mathcal{E}(n_0) = \mathcal{E}$, $(A, \{A, \top :>\})$ is choosable for *any* $>$. Thus we label n_0 with A and add $\{A, \top : a?\bar{a}\}$ to P, where "?" is some arbitrary order ($<$ or $>$) over $\{a, \bar{a}\}$. Since we are working in the UP-case the algorithm then calls $\mathsf{generateUncondLabel}(\mathcal{E}, A) = \{a, \bar{a}\}$ and generates a single edge from n_0 labelled with $\{a, \bar{a}\}$ and leading to a new unlabelled node n_1. The examples $\mathcal{E}(n_1)$ corresponding to the next node will be just $\{(\alpha, \beta) \in \mathcal{E} \mid \alpha(A) = \beta(A)\} = \mathcal{E}$ (i.e., no examples in $\mathcal{E}$ are removed). [5] At the next node n_1, with A now taken care of, we check if $(B, \textit{newRules})$ is UP-choosable w.r.t. $\mathcal{E}(n_1), \{A\}, P$. We see that it is not UP-choosable, owing to the opposing preferences over B exhibited for instance in examples 1,2 of $\mathcal{E}$. However $(C, \{C, \top : c > \bar{c}\})$ *is* UP-choosable, thus the algorithm labels n_1 with C and adds $C, \top : c > \bar{c}$ to P. At the next node n_2 we have $\mathcal{E}(n_2) = \{(\alpha, \beta) \mid \alpha(\{A, C\}) = \beta(\{A, C\})\} = \{1, 2, 3, 4\}$. But the only remaining attribute B is not UP-choosable w.r.t. $\mathcal{E}(n_2), \{A, C\}, P$ (because for instance we still have $1, 2 \in \mathcal{E}(n_2)$). Thus the sub-algorithm $\mathsf{chooseAttribute}(\mathcal{E}(n_2), \{A, C\}, P)$ returns FAILURE and so does GenerateLPStructure in this case (see the left-hand side of Fig. 3). Hence there is no UP&I structure consistent with $\mathcal{E}$.

However the algorithm *does* successfully return a *CP-UI* structure. This is because, at node n_1, even though $(B, \textit{newRules})$is not UP-choosable w.r.t. $\mathcal{E}(n_1), \mathsf{Anc}(n_1), P$ for any appropriate choice of *newRules* (i.e., of the form $B, \top :>$ in the UP-case), it *is* CP-choosable. Recall that to be CP-choosable, *newRules* must contain a rule $B, u :>$ for each $u \in \underline{n}_1 = \{a, \bar{a}\}$, and in this case we may take $\textit{newRules} = \{B, a : b > \bar{b}, B, \bar{a} : \bar{b} > b\}$. After this, since there is no $(\alpha, \beta) \in \mathcal{E}(n_1)$ such that $\alpha(B) = \beta(B)$, $\mathsf{generateUncondLabel}(\mathcal{E}(n_1), B)$ generates no labels and the algorithm terminates with the CP-UI structure on the right-hand side of Fig. 3.

Example 3. Consider the following examples:

$$1.\ (ab\bar{c}, abc)\ \ 2.\ (abc, a\bar{b}c)\ \ 3.\ (ab\bar{c}, a\bar{b}\bar{c})\ \ 4.\ (\bar{a}\bar{b}c, \bar{a}bc)\ \ 5.\ (\bar{a}bc, \bar{a}\bar{b}\bar{c})$$

We will now use the algorithm to check if there is a CP&I structure consistent with these examples. We start at the root node n_0, and check whether $(A, \textit{newRules})$ is CP-choosable

[5] Note in fact A is really a completely uninformative choice here, since it does not decide any of the examples. A sensible heuristic for the algorithm - at least in the UP case - would be to disallow choosing any attribute X such that $\alpha(X) = \beta(X)$ for all examples. Such heuristics will be addressed in future work.

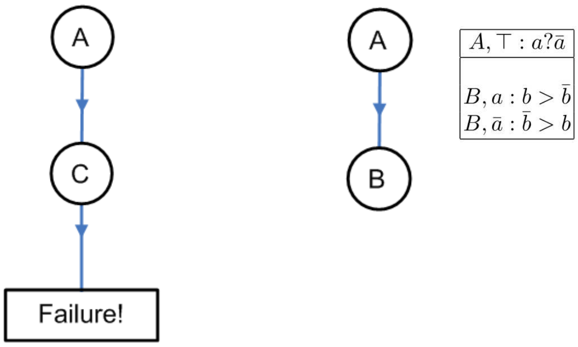

Fig. 3. Output structures for Example 2. *Left:* The output is failure for UP&I structures. *Right:* The output CP-UI structure.

w.r.t. $\mathcal{E}, \emptyset, \emptyset$. As in the previous example, since $\alpha(A) = \beta(A)$ for all $(\alpha, \beta) \in \mathcal{E}$, we may label n_0 with A, and add preference rule $A, \top : a?\bar{a}$ to P, where "?" is some arbitrary preference between $a, \bar{a}$. Since we are now in the CI-case, algorithm $\mathsf{generateCondLabels}(\mathcal{E}, A)$ is called, which generates an edge-label for each value x of A such that $\alpha(A) = \beta(A) = x$ for some $(\alpha, \beta) \in \mathcal{E}$, in this case both a (see, e.g., example 1 in $\mathcal{E}$) and $\bar{a}$ (see, e.g., example 4). Thus two edges from n_0 are created, labelled with $a, \bar{a}$ resp., leading to two new unlabelled nodes n_1 and m_1.

Following the right-hand branch leading to m_1 first (see Fig. 4), we have $\mathcal{E}(m_1) = \{(\alpha, \beta) \in \mathcal{E} \mid \alpha(A) = \beta(A) = \bar{a}\} = \{4, 5\}$. Here we first check if $(B, \mathit{newRules})$ is CP-choosable w.r.t. $\mathcal{E}(m_1), \{A\}, P$. By definition of CP-choosable *newRules* must be of the form $\{B, \bar{a} :>\}$. However due to the opposing preferences on their restriction to B exhibited by 4,5, we see there is no possible choice for $>$ here. Thus we have to consider C instead. Here we see $(C, \{C, \bar{a} : c > \bar{c}\})$ is CP-choosable, thus m_1 is labelled with C, and $C, \bar{a} : c > \bar{c}$ is added to P. Since $\mathsf{generateCondLabel}(\mathcal{E}(m_1), C) = \emptyset$, no new nodes are created on this branch.

Now, moving back to n_0 and following the left-hand branch to node n_1, we have $\mathcal{E}(n_1) = \{(\alpha, \beta) \in \mathcal{E} \mid \alpha(A) = \beta(A) = a\} = \{1, 2, 3\}$. Checking B for CP-choosability first, we see $(B, \{B, a : b > \bar{b}\})$ is CP-choosable w.r.t. $\mathcal{E}(n_1), \{A\}, P$, thus n_1 is labelled with B and $B, a : b > \bar{b}$ added to P; $\mathsf{generateCondLabel}(\mathcal{E}(n_1), B) = \{\{b\}\}$, thus one edge is generated, labelled with b, leading to new node n_2 with $\mathcal{E}(n_2) = \{(\alpha, \beta) \in \mathcal{E} \mid \alpha(\{A, B\}) = \beta(\{A, B\}) = ab\} = \{1\}$. For the last remaining attribute C on this branch we have $(C, \{C, ab : \bar{c} > c\})$ is CP-choosable w.r.t. $\mathcal{E}(n_2), \{A, B\}, P$. Thus the algorithm successfully terminates here, labelling n_2 with C and adding $C, ab : \bar{c} > c$ to P. The constructed CP&I structure in Fig. 4 is thus consistent with $\mathcal{E}$.

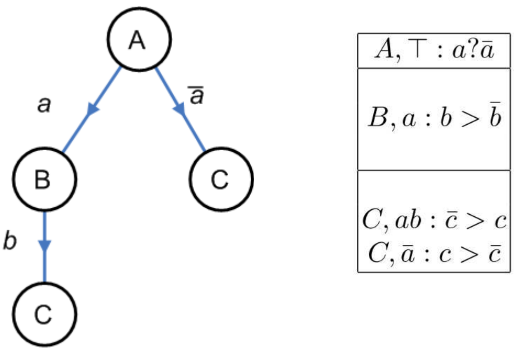

Fig. 4. Output CP&I structure for Example 3.

5.3 Complexity of model identification

In this section, we prove that the greedy algorithm does indeed return a LP-structure of the right type when one exists.

We first prove that the algorithm is correct. The table in Fig. 5 gives the parameters for the greedy algorithm that solve five learning problems. In fact, the only problem that cannot be solved with this algorithm, as will be shown below, is the learning of a UP-CI structure without initial knwoldge of the preferences.

Proposition 6. *Using the right type of labels and the right choosability condtion and the right initial preference table, if the algorithm does not stop with FAILURE when called on a given set $\mathcal{E}$ of examples, then it returns a structure of the expected type, as described in the table of Fig. 5, consistent with $\mathcal{E}$, if such a structure exists*

learning problem	choosability	labels	initial P	structure type
CP&I	CP-choosable	conditional	$\emptyset$	CP&I
CP-UI	CP-choosable	uncond.	$\emptyset$	CP-UI
UP&I	UP-choosable	uncond	$\emptyset$	UP&I
FP-CI	UP-choosable	conditional	1 rule/attr.	UP-CI
FP-UI	UP-choosable	uncond	1 rule/attr.	UP-CI

Fig. 5. Parameters of the greedy algorithm for five learning problems

Proof. Note first that the while-loop is entered at least once, since when the root of T is created it is not labelled, thus when the algorithm finishes without failure P is not ambiguous. Furthermore in this case, every node is labelled with an attribute, and T is a attribute-importance tree. When a node n is examined in the while loop, if $(X, \mathit{newRules})$ is chosen and if $(\alpha, \beta) \in \mathcal{E}(n)$ then either

$\alpha(X) \neq \beta(X)$ and there is a rule $X, v :>$ in $P \cup \mathit{newRules}$ that covers (α, β) at n; in this case, (α, β) is decided at n and, since no rule is ever retracted from P, in the end $\alpha >_\sigma \beta$; or

$\alpha(X) = \beta(X)$ in which case there is a unique child n' of n such that $(\alpha, \beta) \in \mathcal{E}(n')$ (the one corresponding to the edge labelled with a subset of $\underline{X}$ that contains $\alpha(X)$).

The latter case shows that when the algorithm finishes without failure, every example is decided at some node of the tree, and the first case shows that the final structure is consistent with each example. It is easy to check that the algorithm return a structure of a particular type when it uses the right type of helper function.

We now turn to the completness of the algorithm: we need to show that if there exists a structure of a given type consistent with the set of examples, then the algorithm, using the right type of helper functions, will not return $\mathsf{FAILURE}$: so we need to prove that at every stage there is an attribute that can be chosen with an appropriate set of rules.

We first note that every attribute that appears at some node in some LP-structure is choosable at that node in the following sense:

Proposition 7. *Let (T, P) be some partial LP-structure – i.e. such that T contains some unlabelled leaves – and let n be some unlabelled node of T. If there exists some (complete) LP-structure (T', P') that extends (T, P), such that X is the label of n and such that R is the set of rules of P' that apply at n, then (X, R) is choosable with respect to $\mathcal{E}(n)$, $\mathsf{Anc}(n)$, P.*

Proof. Since $P \cup R \subseteq P'$, it is not ambiguous; moreover, if $(\alpha, \beta) \in \mathcal{E}(n)$ and $\alpha(X) \neq \beta(X)$, then $\alpha >_{(T',P')} \beta$ because the pair is decided at n in (T', P') and the structure is consistent with $\mathcal{E}$, so $\alpha(X) > \beta(X)$, so there is some rule $X, u :>$ that applies to (α, β) at n, in which case the rule is in R by definition of R.

There remains to prove that the algorithm does not run into a deadend; that, whatever the chosen attribute and rule(s) at every stage, it will be possible to finish the tree (under the assumption that a structure consistent with the examples does exist).

Lemma 2. *Let $\sigma = (T, P)$ be some CP&I-LP-structure consistent with $\mathcal{E}$ of examples. Let n be some node of σ, let T_n be the subtree of T rooted at n. Let X be some attribute, and let P^n be the set of rules of P that apply at nodes of T_n. Suppose that the pair $(X, \{X, u :>\})$ is choosable with respect to $\mathcal{E}(n)$, $\mathsf{Anc}(n)$, $P - P^n$, were $u \in \underline{\mathsf{Anc}(n)}$. Then there exists a CP&I-LP-structure $\sigma' = (T', P')$ consistent with $\mathcal{E}$, with T' identical to T except that T_n is replaced in T' with a new subtree whose root m is labelled with X, and with P' containing $(P - P^n) \cup \{X, u :>\}$.*

Proof. Let $P' = (P - P^n) \cup \{X, u :>\} \cup \{Z, xv :>' \mid Z, v :>' \in P^n, Z \neq X, x \in \underline{X}\}$; and let T' be the tree obtained as follows: first make one copy of T_n for each value of $\underline{X}$, create a node m labelled with X and attach the copies of T_n to it, labelling each edge between m and a copy of T_n with a value of $\underline{X}$, and replace T_n with the new subtree T_m starting at m; second, remove from T_m each node labelled with X, keeping only the branch below such node that corresponds to the value of X on the current branch (the transformation is depicted on Fig. 6). We will prove that $\sigma' = (T', P')$ is CP&I-LP-structure that is consistent with $\mathcal{E}$.

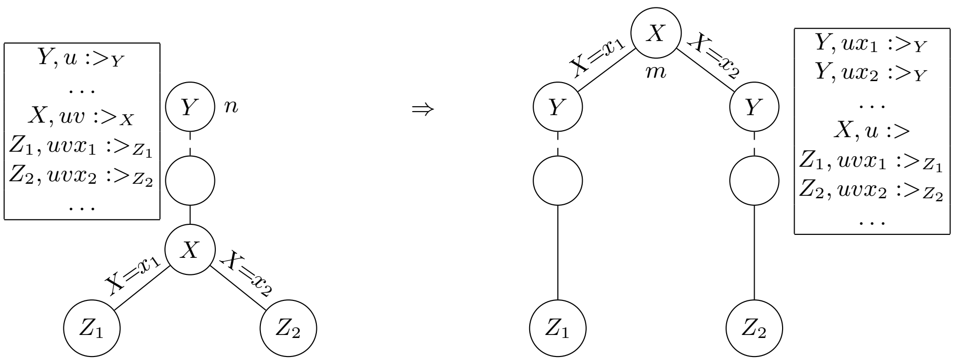

Fig. 6. Adding a choosable attribute at the top of a CP&I-LP-structure

Since $(X, \{X, u :>\})$ is choosable with respect to $\mathcal{E}(n)$, $\mathsf{Anc}(n)$ and $P - P^n$, P' is not ambiguous. By construction, σ' is a CP&I-LP-structure. In order to prove that it is consistent with $\mathcal{E}$, consider an example $(\alpha, \beta) \in \mathcal{E}$. Obviously, if it is decided in σ at a node that is not in T_n, then it is still decided in the same way in σ'. Suppose now that (α, β) is decided by node n' of T_n. Then $\alpha(\mathsf{Anc}_\sigma(m)) = \beta(\mathsf{Anc}_\sigma(m))$ since $\mathsf{Anc}_{\sigma'}(n') \supseteq \mathsf{Anc}_{\sigma'}(n) = \mathsf{Anc}_\sigma(m)$. If X labels n', and if $X, u' :>'$ is the rule that applies to n' in T (with $\{u'\} = \underline{\mathsf{Anc}(n')}$), then $\alpha(X) \neq \beta(X)$, and $\alpha(X) > \beta(X)$ since $(X, \{X, u :>\})$ is choosable with respect to $\mathcal{E}(n)$, hence $\alpha >_\sigma \beta$. Lastly, consider the case where n' is labelled with $(Z, >')$, where $Z \neq X$: if $\alpha(X) \neq \beta(X)$, then $\alpha(X) > \beta(X)$ since $(X, \{X, u :>\})$ is choosable for m, thus $\alpha >_\sigma \beta$; if $\alpha(X) = \beta(X)$, then (α, β) is decided in σ' at the node corresponding to n' in the branch of T_m that corresponds to the value given to X by both α and β, and since the node is the same as in σ that is consistent with $\mathcal{E}$, we have $\alpha >_\sigma \beta$.

Proposition 8. *Using the right type of labels and the right choosability condtion and the right initial preference table, the algorithm returns, when called on a given set $\mathcal{E}$ of examples, a structure of the expected type, as described in the table on Fig. 5, consistent with $\mathcal{E}$, if such a structure exists.*

Proof. Prop. 6 shows that the structure has the right properties if the algorithm does not stop with FAILURE. For completeness, Lemma 2 proves that an invariant of the while loop of the algorithm is that there is always a CP&I LP-structure (T, P) that extends the current one and that is consistent with the set of examples, and Prop. 7 proves that in this case there is always a CP-choosable attribute with associated conditional rules.

CP&I Lemma 2 proves that an invariant of the while loop of the algorithm is that there is always a CP&I LP-structure (T, P) and that is consistent with the set of examples, and Prop. 7 proves that in this case there is always a CP-choosable attribute with associated conditional rules.

CP-UI A construction similar to that used in the proof of Lemma 2 will work: in fact, a CP-UI LP-structure is equivalent to a CP&I one where the branches in the tree are all identical, and we can modify at once all branches of this tree below the level of a given node.

UP&I Similar to previous case, except that here there is only one rule associated to each attribute.

FP-CI Similar to the CP&I case, except that here we use UP-choosable pair (X, R) where R contains only one rule that is already in the preference table: because it is already there, we know it can be used in all branches of the modified tree, even the unmodified ones.

FP-UI Similar to the UP&I case.

Corollary 1. *The problems of deciding if there exists a LP-structure of a given class consistent with a given set of examples over binary attributes have the following complexities:*

	FLP	*ULP*	*CLP*
UI	***P***(*Dombi* et al. , *2007*)	***P***	***P***
CI	***P***	***NP**-complete*	***P***

Proof. Previous results show that the greedy algorithm is correct and complete for the CP&I, CP-UI, FP-CI, FP-UI and UP&I cases. There remains to prove that it runs in polynomial time. The number L of leaves ot the tree cannot exceed $|\mathcal{E}|$; and each leaf cannot be at depth greater than n, so it cannot have more than $n-1$ ancestors, so the tree cannot have more than $(n-1) \times |\mathcal{E}|$ nodes, so there can be no more that $(n-1) \times |\mathcal{E}|$ iterations of the while loop. Every step of this loop except (2b) is executed in linear time. In order to choose an attribute, we can, for each remaining attribute X, consider the relation $\{(\alpha(X), \beta(X)) \mid (\alpha, \beta) \in \mathcal{E}(n)\}$ on $\underline{X}$: we can check in polynomial time if it has cycles, and, if not, extend it to a total strict relation over $\underline{X}$. This proves membership in **P** for these cases.

Membership in **NP** for the UP-CI case comes from the following nondeterministic algorithm:

1. guess a set of unconditional local preference rules P;
2. check whether there exists a attribute importance tree T such that (T, P) is consistant with $\mathcal{E}$.

Since the size of P is linear in the number of attributes and model identifiability for FLP-CI is in **P**, model identifiability for ULP-CI is in **NP**.

Hardness comes from the following reduction from WEAK SEPARABILITY – the problem of checking if there is a CP-net without dependencies *weakly consistent* with a given set of examples – shown to be **NP**-complete by Lang & Mengin (2009). More precisely, a set of examples $\mathcal{E}$ is weakly separable if and only if there exists a (non ambiguous) set of unconditional preference rules that contains, for every $(\alpha, \beta) \in \mathcal{E}$, a rule $X, \top :>$ such that $\alpha(X) > \beta(X)$. Let $\mathcal{E} = \{(\alpha_1, \beta_1), \ldots, (\alpha_m, \beta_m)\}$ be a set of examples, built on a set of attributes $\{X_1, \ldots, X_n\}$. We introduce m new attributes $P_1, \ldots, P_m$ and map $\mathcal{E}$ to the following set of examples $\mathcal{E}'$ built as follows: for every $e_i = (\alpha_i, \beta_i)$ in $\mathcal{E}$ we create the example

$$e'_i = (\alpha_i \overline{p_1} \ldots \overline{p_{i-1}} p_i \overline{p_{i+1}} \ldots \overline{p_m} \, , \, \beta_i \overline{p_1} \ldots \overline{p_{i-1}} p_i \overline{p_{i+1}} \ldots \overline{p_m}).$$

We claim that there is some UP-CI LP-structure consistent with $\mathcal{E}' = \{e'_i \mid e_i \in \mathcal{E}\}$ if and only if $\mathcal{E}$ is weakly separable.

Assume first that $\mathcal{E}$ is weakly separable: there exists a set of unconditional preference rules P^* that is weakly compatible (in the sense of Lang & Mengin (2009)) with $\mathcal{E}$. For every i, since P is weakly compatible with (α_i, β_i), there exists a attribute $X_{k(i)}$ such that either (a) P contains $X_{k(i)}, \top : x_{k(i)} > \overline{x_{k(i)}}$, $\alpha_i(X_{k(i)}) = x_{k(i)}$ and $\beta_i(X_{k(i)}) = \overline{x_{k(i)}}$ or (b) P contains $X_{k(i)}, \top : \overline{x_{k(i)}} > x_{k(i)}$, $\alpha_i(X_{k(i)}) = \overline{x_{k(i)}}$ and $\beta_i(X_{k(i)}) = x_{k(i)}$. Let $P = P^* \cup \{P_i, \top : p_i > \overline{p_i} \mid 1 \leq i \leq m\}$, and consider the following attribute importance tree T:

- all branches of T start with nodes labelled with$P_1, \ldots, P_m$ at the m first levels;
- for every branch in which exactly one of the P's is true, there is exactly one example $(\alpha_i, \beta_i) \in \mathcal{E}'(n)$ where n is the node at the m-th level: the remaining importance order (after the P_j's) starts with $X_{k(i)}$, and contains whatever afterwards.
- in all other branches, choose any priority order after the P_j's.

It is easy to check that (T, P) is consistent with $\mathcal{E}'$: for every $1 \leq i \leq m$ $e'_i = (\alpha'_i, \beta'_i)$, where $\alpha'_i = \alpha_i \overline{p_1} \ldots \overline{p_{i-1}} p_i \overline{p_{i+1}} \ldots \overline{p_m}$ and $\beta'_i = \beta_i \overline{p_1} \ldots \overline{p_{i-1}} p_i \overline{p_{i+1}} \ldots \overline{p_m}$; α'_i and β'_i coincide on all P_j's, and disagree on the next most important attribute, namely $X_{k(i)}$. If we are in case (a) then the preferred value for $X_{k(i)}$ is $x_{k(i)}$, α'_i assigns $X_{k(i)}$ to $x_{k(i)}$ and β'_i assigns it to $\overline{x_{k(i)}}$, therefore $\alpha'_i >_{(T,P)} \beta'_i$. Case (b) is symmetric.

Conversely, assume there is some CP-UI LP-structure (T, P') consistent with $\mathcal{E}'$. Let P be the restriction of P' to the attributes $X_1, \ldots, X_n$. We claim that (T, P) is weakly compatible with $\mathcal{E}$. If not, then there would exist an example e_i in $\mathcal{E}$ such that $P \models \beta_i > \alpha_i$. Consider $\alpha'_i = \alpha_i \overline{p_1} \ldots \overline{p_{i-1}} p_i \overline{p_{i+1}} \ldots \overline{p_m}$ and $\beta'_i = \beta_i \overline{p_1} \ldots \overline{p_{i-1}} p_i \overline{p_{i+1}} \ldots \overline{p_m}$. The most important attribute in T on which they disagree is a attribute X_j such that α_i gives the preferred value to X_j and β_j the dispreferred (because $P \models \beta_i > \alpha_i$). Therefore we would have $\beta'_i >_{(T,P)} \alpha'_i$, a contradiction. From this we conclude that P is weakly compatible with $\mathcal{E}'$, thus that $\mathcal{E}'$ is weakly separable.

5.4 Complexity of model approximation

In practice, a general problem in machine learning is that there is often no structure of a given type that is consistent with all the examples at the same time. It is then interesting to find a structure that is consistent with the most examples. Schmitt & Martignon (2006) have shown that finding a UI&LP-structure, with a fixed set of local preferences, that satisfies as many examples from a given set as possible, is **NP**-complete, in the case where all attributes are binary. We extend these results, still when attributes are all binary. In this case, we will denote by x and $\overline{x}$ the two values of attribute X.

Proposition 9. *The complexities of finding a LP-structure in a given class , which wrongly classifies at most k examples of a given set $\mathcal{E}$ of examples over* binary *attributes, for a given k, are as follows:*

	FLP	*ULP*	*CLP*
UI	***NP**-completeSchmitt & Martignon (2006)*	***NP**-complete*	***NP**-hard*
CI	***NP**-complete*	***NP**-complete*	***NP**-complete*

Proof. These problems are in NP because in each case a witness is the LP-structure that has the right property, and such a structure need not have more nodes than there are examples. For the UP-CI case, the problem is already NP-complete for $k = 0$, so it is NP-hard. NP-hardness of the other cases follow from the case proved by Schmitt & Martignon (2006).

NP-hardness for the UP&I case comes from from the following reduction from FP-UI: given a set of examples $\mathcal{E}$, and a set P of unconditional rules, one $X, \top : x_1 > x_2$ for each attribute X, where $x_1, x_2 \in \{x, \overline{x}\}$, let $\mathcal{E}'$ contain $\mathcal{E}$ and $k+1$ copies of the example (x_1u, x_2u) for each attribute X, where u is some assignment to the attributes in $\mathcal{A} - \{X\}$. Suppose that there is a UP&I structure $\sigma = (T, P)$ that makes less than k errors on $\mathcal{E}$, clearly σ makes less than k errors on $\mathcal{E}'$. Conversely, if there is some structure that makes less than k errors on $\mathcal{E}'$, then it cannot make any error on the examples that where not in $\mathcal{E}$ (it this were the case, it would make $k+1$ times the same error), so its preference table is P, and it makes less than k errors on $\mathcal{E}$.

NP-hardness for the CP-UI and CP&I cases follows from the following reduction from UP&I. Given a set of examples $\mathcal{E}$, for each $(\alpha, \beta) \in \mathcal{E}$, let (α', β') be defined as follows: for each attribute X, if $\alpha(X) = \beta(X) = x$ then set $\alpha'(X) = \beta'(X) = \overline{x}$; otherwise, $(\alpha'(X), \beta'(X)) = (\alpha(X), \beta(X))$. Let $\mathcal{E}'$ be the set of the transformed examples. Note that a UP&I LP-structure orders in the same way (α, β) and (α', β'), so it makes k errors on $\mathcal{E}$ if and only if it also makes k errors on $\mathcal{E}'$. Suppose now that we learn a CP-UI or a CP&I LP-structure that makes no more than k errors on $\mathcal{E}'$: all examples of $\mathcal{E}'$ are decided in the same "branch", the one that assigns $\overline{x}$ to every attribute X. So we can easily obtain a UP&I structure that makes no more than k errors on $\mathcal{E}$. And conversely, if there is a UP&I structure that makes no more than k errors on $\mathcal{E}$, then by multiplying the unconditional preference rules to make conditional ones for each partial assignment at every node, we obtain a CP-UI structure that makes no more than k errors on $\mathcal{E}'$; by multiplicating further the branches, we also obtain a CP&I structure that has the same property.

NP-hardness for the FP-CI case follows from a reduction from CP&I. Given a set of examples $\mathcal{E}$ over set of attributes $\mathcal{A}$, define a set of examples $\mathcal{E}'$ over a set of attributes $\mathcal{A}'$ that contains $\mathcal{A}$ and for each $X \in \mathcal{A}$ a new attribute X' with the same domain. For $(\alpha, \beta) \in \mathcal{E}$, define $(\alpha', \beta') = (\alpha\overline{\alpha}, \beta\overline{\beta})$ over $\mathcal{A}'$, where $\alpha\overline{\alpha}$ denote the concatenation of α with the vector $\overline{\alpha}$ of the opposite values. Consider the preference table P' that contains $X, \top : x > \overline{x}$ for every $X \in \mathcal{A}$ and $X', \top : x' > \overline{x'}$ for every $X' \in \mathcal{A}'$. If there is a CP&I LP-structure (T, P) that make k errors on $\mathcal{E}$, let T' be the tree obtained by replacing in T each node n labelled with a attribute X such that P contains $X, \underline{n} : \overline{x} > x$ with a node labelled with X': then (T', P') makes k errors on $\mathcal{E}'$, and any algorithm able to check if there is a UP-CI structure with table P' that makes no more than k errors on $\mathcal{E}'$ will answer yes. Conversely, if such an algorithm answers yes, that is, if there is T' such that (T', P') make no more than k errors on $\mathcal{E}'$, then it can be easily transformed into a CP&I LP-structure that makes the corresponding mistakes on $\mathcal{E}$: replace in T' every node labelled with $X' \in \mathcal{A}'$ with a node labelled with the corresponding $X \in \mathcal{A}$, and put in P conditional rules $X, n : x > \overline{x}$ or $X, n : \overline{x} > x$ depending on whether node n was labelled with X or X' in T'.

6 Conclusion and future work

We have proposed a general, lexicographic type of models for representing a large family of preference relations. We have defined six interesting classes of models where the attribute importance as well as the local preferences can be conditional, or not. Two of these classes correspond to the usual unconditional lexicographic orderings, and to a variant of Wilson's "Pre-Order Search Trees" (or POST) (2006). Interestingly, classes where preferences are conditional have an exponentional VC dimension.

We have calculated the cardinality of five of these six classes, and proved that the communication complexity for each class is not greater than the log of this cardinality, thereby generalizing a previous result by Dombi *et al.* (2007).

As for passive learning, we have proved that a greedy algorithm like the ones proposed by Schmitt & Martignon (2006); Dombi *et al.* (2007) for the class of unconditional preferences can identify a model in another four classes, thereby showing that the model identification problem is polynomial for these classes. We have also proved that the problem is NP-complete for the class of models with conditional attribute importance but unconditional local preferences. On the other hand, finding a model that minimizes the number of mistakes turns out to be NP-complete in all cases.

Our LP-structures are closely connected to decision trees. In fact, one can prove that the problem of learning a decision tree consistent with a set of examples can be reduced to a problem of learning a CP-CI LP structure. There remains to see if CP-CI structures can be as efficiently learnt in practice as decision trees. As future work, we intend to test our algorithms, with appropriate heuristics to guide the choice of variables a each stage. A possible heuristics would be the mistake rate if some unconditional structure is built below a given node (which can be very quickly done). Another interesting aspect would be to study mixtures of conditional and unconditional structures, with e.g. the first two levels of the structure being conditional ones, the remaining ones being unconditional (since it is well-known that learning decision trees with only few levels can be as good as learning trees with more levels).

References

Booth, Richard, Chevaleyre, Yann, Lang, Jérôme, Mengin, Jérôme, & Sombattheera, Chattrakul. 2009. *Learning various classes of models of lexicographic orderings*. Tech. rept. Institut de Recherche en Informatique de Toulouse.

Boutilier, Craig (ed). 2009. *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*.

Dimopoulos, Yannis, Michael, Loizos, & Athienitou, Fani. 2009. Ceteris Paribus Preference Elicitation with Predictive Guarantees. *In:* Boutilier (2009).

Dombi, József, Imreh, Csanád, & Vincze, Nándor. 2007. Learning Lexicographic Orders. *European Journal of Operational Research*, **183**, 748–756.

Koriche, Frédéric, & Zanuttini, Bruno. 2009. Learning conditional preference networks with queries. *In:* Boutilier (2009).

Lang, Jérôme, & Mengin, Jérôme. 2009. The complexity of learning separable ceteris paribus preferences. *In:* Boutilier (2009).

Schmitt, Michael, & Martignon, Laura. 2006. On the Complexity of Learning Lexicographic Strategies. *Journal of Machine Learning Research*, **7**, 55–83.

Wilson, Nic. 2006. An Effcient Upper Approximation for Conditional Preference. *In:* Brewka, Gerhard, Coradeschi, S., Perini, A., & Traverso, P. (eds), *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI 2006)*.

Yaman, Fusun, Walsh, Thomas J., Littman, Michael L., & desJardins, Marie. 2008. Democratic Approximation of Lexicographic Preference Models. *In: Proceedings of the 25th International Conference on Machine Learning (ICML'08)*.