Learning various classes of models of lexicographic orderings

Richard Booth, Mahasarakham University, Thailand
Yann Chevaleyre, LAMSADE, Université Paris-Dauphine
Jérôme Lang, LAMSADE, Université Paris-Dauphine
Jérôme Mengin, IRIT, Université de Toulouse
Chattrakul Sombattheera, Mahasarakham University, Thailand

Introduction

Topic: learn to order objects of a combinatorial domain

E.g. computers, described by
Type: *d*esktop or *l*aptop
Color: *y*ellow or *b*lack
Dvd-unit: *r*eader or *w*riter . . .
Recommender system : learn how a user orders these objects, in order to suggest the "best" ones among those that are available / the user can afford.

If n variables, domains of m values : m^n objects, $!m^n$ orderings
$\Rightarrow$ need compact representation of the orderings :

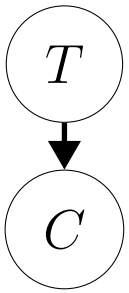
- local preferences on each attribute
- extra structure on the set of variables to "aggregate" to global preferences

Introduction

Lexicographic orderings :

local preferences over the domains of each variable

$+$ *importance ordering* of the variables



$l \succ d$

$y \succ b$

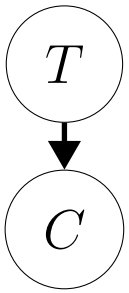
- Type is more important than Colour
- Prefer laptop to desktop
- Prefer yellow to black

Introduction

Lexicographic orderings :

local preferences over the domains of each variable

$+$ *importance ordering* of the variables



$l \succ d$

$y \succ b$

$\mathbf{l}b \succ \mathbf{d}y$ (decided at node T)

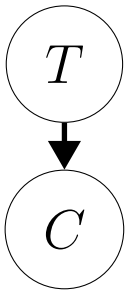
$l\mathbf{y} \succ l\mathbf{b}$ (decided at node C)

Introduction

Lexicographic orderings :

local preferences over the domains of each variable

$+$ *importance ordering* of the variables



$l \succ d$

$y \succ b$

$+$ comparisons in linear time

$+$ learning in polynomial time [SM06, DIV07]

$-$ very weak expressive power:

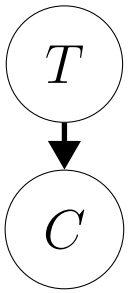
"prefer yellow for laptops, black for desktops"

Introduction

Lexicographic orderings :

local preferences over the domains of each variable

$+$ *importance ordering* of the variables



$l \succ d$

$y \succ b$

$+$ comparisons in linear time

$+$ learning in polynomial time [SM06, DIV07]

$-$ very weak expressive power:

"prefer yellow for laptops, black for desktops"

Conditional Preference Networks (CP-nets) :

conditional local preferences (dependency graph)

e.g.: $l : y \succ b$ (for laptops: yellow pref. to black) $d : b \succ y$ $l \succ d$

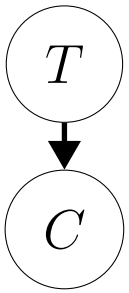
$+$ *ceteris paribus* comparisons: $ly \succ lb \succ db \succ dy$

Introduction

Lexicographic orderings :

local preferences over the domains of each variable

$+$ *importance ordering* of the variables



$l \succ d$

$y \succ b$

$+$ comparisons in linear time

$+$ learning in polynomial time [SM06, DIV07]

$-$ very weak expressive power:
"prefer yellow for laptops, black for desktops"

Conditional Preference Networks (CP-nets) :

conditional local preferences (dependency graph)

e.g.: $l : y \succ b$ (for laptops: yellow pref. to black) $d : b \succ y$ $l \succ d$

$+$ *ceteris paribus* comparisons: $ly \succ lb \succ db \succ dy$

$+$ very expressive

$-$ comparisons difficult (NP-complete)

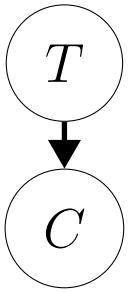
$-$ hard to learn [session on CP-net learning at IJCAI'09]

Introduction

Lexicographic orderings :

local preferences over the domains of each variable

+ *importance ordering* of the variables



$l \succ d$

$y \succ b$

+ comparisons in linear time

+ learning in polynomial time [SM06, DIV07]

− very weak expressive power:

"prefer yellow for laptops, black for desktops"

Conditional Preference Networks (CP-nets) :

conditional local preferences (dependency graph)

e.g.: $l : y \succ b$ (for laptops: yellow pref. to black) $d : b \succ y$ $l \succ d$

+ *ceteris paribus* comparisons: $ly \succ lb \succ db \succ dy$

+ very expressive

− comparisons difficult (NP-complete)

− hard to learn [session on CP-net learning at IJCAI'09]

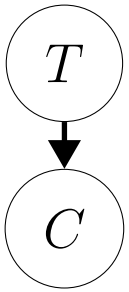
(easy classes of CP-nets / examples, incomplete algorithms)

Introduction

Lexicographic orderings :

local preferences over the domains of each variable

+ *importance ordering* of the variables



$l \succ d$

$y \succ b$

+ comparisons in linear time

+ learning in polynomial time [SM06, DIV07]

− very weak expressive power:
"prefer yellow for laptops, black for desktops"

Conditional Preference Networks (CP-nets) :

conditional local preferences (dependency graph)

e.g.: $l : y \succ b$ (for laptops: yellow pref. to black) $d : b \succ y$ $l \succ d$

+ *ceteris paribus* comparisons: $ly \succ lb \succ db \succ dy$

+ very expressive

− comparisons difficult (NP-complete)

− hard to learn [session on CP-net learning at IJCAI'09]

$\Rightarrow$ **find something in between the two formalisms**

Introduction

Contribution of this paper:

it is possible to add conditionality in lexicographic prefence models without increasing the complexity of reasoning / learning

Learning unconditional lexicographic preferences

Sample complexity: VC dim $= n$ (when n variables, all binary)

Learning unconditional lexicographic preferences

Sample complexity: VC dim $= n$ (when n variables, all binary)

Active learning: a learner asks "user" queries of the form "What is preferred between ly and bd ?"
Goal : identify preference model of the user

$\Rightarrow$ If local pref. fixed, need $\log(!n)$ queries (worst case) [DIV07]
$\Rightarrow$ If local pref. to be learnt, need $n + \log(!n)$ queries

Learning unconditional lexicographic preferences

Sample complexity: VC dim $= n$ (when n variables, all binary)

Active learning: a learner asks "user" queries of the form "What is preferred between ly and bd ?"
Goal : identify preference model of the user

$\Rightarrow$ If local pref. fixed, need $\log(!n)$ queries (worst case) [DIV07]
$\Rightarrow$ If local pref. to be learnt, need $n + \log(!n)$ queries

Passive learning: given set of examples e.g. $\mathcal{E} = \{lb \succ db, \ldots\}$
Goal: output preference struct. consistent with the examples

Learning unconditional lexicographic preferences

Sample complexity: VC dim $= n$ (when n variables, all binary)

Active learning: a learner asks "user" queries of the form "What is preferred between ly and bd ?"
Goal : identify preference model of the user

$\Rightarrow$ If local pref. fixed, need $\log(!n)$ queries (worst case) [DIV07]
$\Rightarrow$ If local pref. to be learnt, need $n + \log(!n)$ queries

Passive learning: given set of examples e.g. $\mathcal{E} = \{lb \succ db, \ldots\}$
Goal: output preference struct. consistent with the examples
Greedy algorithm [DIV07] (return failure if not possible)
$\Rightarrow$ passive learning with fixed local pref. in P [DIV07]
$\Rightarrow$ passive learning with unknown local pref. in P

Learning unconditional lexicographic preferences

Sample complexity: VC dim $= n$ (when n variables, all binary)

Active learning: a learner asks "user" queries of the form "What is preferred between ly and bd ?"
Goal : identify preference model of the user

$\Rightarrow$ If local pref. fixed, need $\log(!n)$ queries (worst case) [DIV07]
$\Rightarrow$ If local pref. to be learnt, need $n + \log(!n)$ queries

Passive learning: given set of examples e.g. $\mathcal{E} = \{lb \succ db, \ldots\}$
Goal: output preference struct. consistent with the examples
Greedy algorithm [DIV07] (return failure if not possible)
$\Rightarrow$ passive learning with fixed local pref. in P [DIV07]
$\Rightarrow$ passive learning with unknown local pref. in P

Model optimization (less than k errors)
$\Rightarrow$ NP-complete with fixed local pref. [SM06]
$\Rightarrow$ NP-complete with unknown local pref.

Learning unconditional lexicographic preferences

Greedy algorithm [DIV07]

1. initialize seq. of var. with empty sequence;
2. while there remains some unused variable:
   (a) choose a variable and local pref. that does not wrongly order the remaining examples
   (b) remove examples ordered with this variable

$$\mathcal{E} = \{lbr \succ dyr,\ \ lyr \succ lbw,\ \ dyw \succ dbr\}$$

Learning unconditional lexicographic preferences

Learning unconditional lexicographic preferences

Greedy algorithm [DIV07]

1. initialize seq. of var. with empty sequence;
2. while there remains some unused variable:
   (a) choose a variable and local pref. that does not wrongly order the remaining examples
   (b) remove examples ordered with this variable

$$\mathcal{E} = \{lbr \succ dyr,\ \ lyr \succ lbw,\ \ dyw \succ dbr\}$$

?

Learning unconditional lexicographic preferences

Greedy algorithm [DIV07]

1. initialize seq. of var. with empty sequence;
2. while there remains some unused variable:
   (a) choose a variable and local pref. that does not wrongly order the remaining examples
   (b) remove examples ordered with this variable

$$\mathcal{E} = \{lbr \succ dyr,\ \ lyr \succ lbw,\ \ dyw \succ dbr\}$$

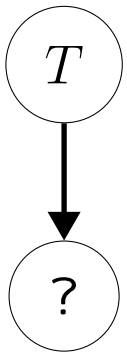
T $\qquad$ $l \succ d$

Learning unconditional lexicographic preferences

Greedy algorithm [DIV07]

1. initialize seq. of var. with empty sequence;
2. while there remains some unused variable:
   (a) choose a variable and local pref. that does not wrongly order the remaining examples
   (b) remove examples ordered with this variable

$$\mathcal{E} = \{ \quad lyr \succ lbw, \;\; dyw \succ dbr \}$$



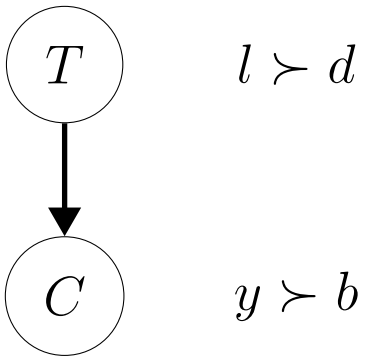
$l \succ d$

Learning unconditional lexicographic preferences

Greedy algorithm [DIV07]

1. initialize seq. of var. with empty sequence;
2. while there remains some unused variable:
   (a) choose a variable and local pref. that does not wrongly order the remaining examples
   (b) remove examples ordered with this variable

$$\mathcal{E} = \{ \quad lyr \succ lbw, \;\; dyw \succ dbr \}$$

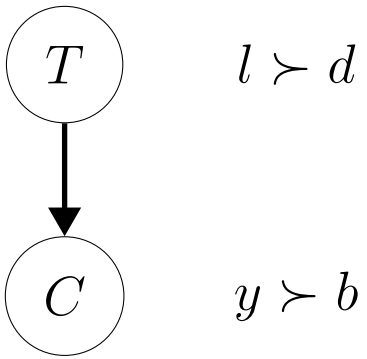


Learning unconditional lexicographic preferences

Greedy algorithm [DIV07]

1. initialize seq. of var. with empty sequence;
2. while there remains some unused variable:
   (a) choose a variable and local pref. that does not wrongly order the remaining examples
   (b) remove examples ordered with this variable

$\mathcal{E} = \{ \quad \}$



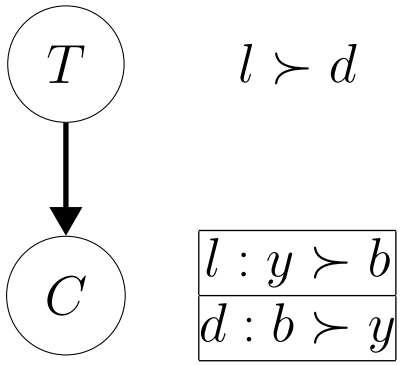
success !

Conditional local pref. / Unconditional variable importance

"I always prefer laptops to desktops"

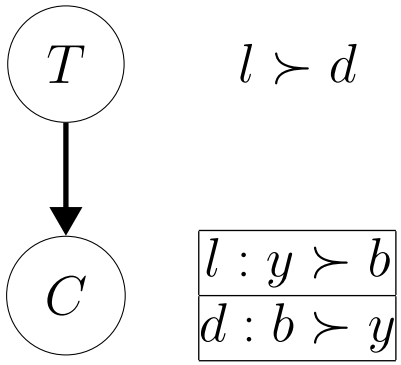
"For desktops, I prefer black to yellow"

"For laptops, I prefer yellow to black"



Conditional local pref. / Unconditional variable importance

"I always prefer laptops to desktops"
"For desktops, I prefer black to yellow"
"For laptops, I prefer yellow to black"



Sample complexity : VC dim $= 2^n - 1$

Active learning : $2^n - 1 + \log(!n)$ queries needed (worst case)

Passive learning : in P (Greedy algorithm still works)

Model optimization : NP-hard

Conditional local pref. / Unconditional variable importance

Greedy algorithm:

$$\mathcal{E} = \{lbr \succ dyr,\ \ lyr \succ lbw,\ \ d\mathbf{b}w \succ d\mathbf{y}r\}$$

Conditional local pref. / Unconditional variable importance

Greedy algorithm:

$$\mathcal{E} = \{lbr \succ dyr,\ \ lyr \succ lbw,\ \ d\mathbf{b}w \succ d\mathbf{y}r\}$$

Conditional local pref. / Unconditional variable importance

Greedy algorithm:

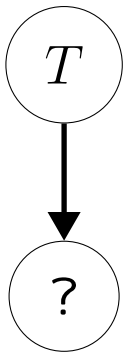
$$\mathcal{E} = \{lbr \succ dyr,\ \ lyr \succ lbw,\ \ d\mathbf{b}w \succ d\mathbf{y}r\}$$

$$\boxed{T} \qquad l \succ d$$

Conditional local pref. / Unconditional variable importance

Greedy algorithm:

$$\mathcal{E} = \{ \quad lyr \succ lbw, \;\; d\mathbf{b}w \succ d\mathbf{y}r \}$$

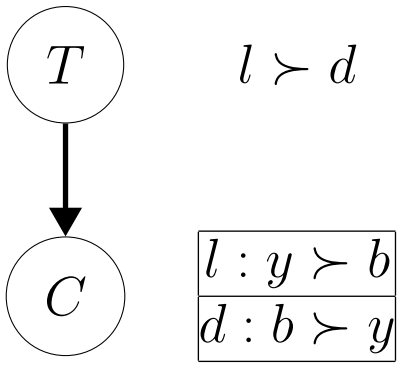


$$l \succ d$$

Conditional local pref. / Unconditional variable importance

Greedy algorithm:

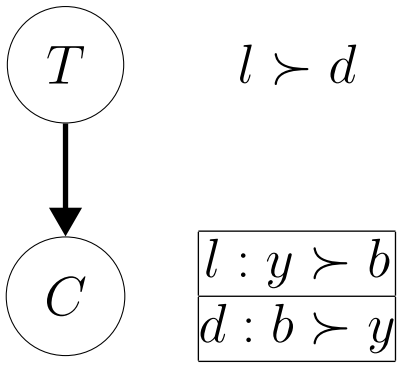
$$\mathcal{E} = \{ \quad lyr \succ lbw, \;\; d\mathbf{b}w \succ d\mathbf{y}r \}$$



Conditional local pref. / Unconditional variable importance

Greedy algorithm:

$\mathcal{E} = \{ \quad \}$

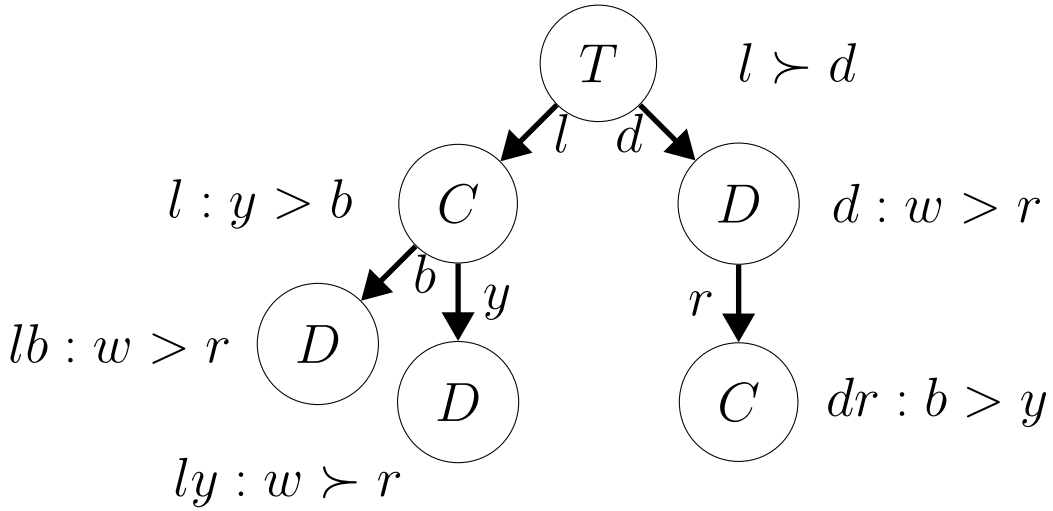


success !

Conditional local pref. & Conditional variable importance

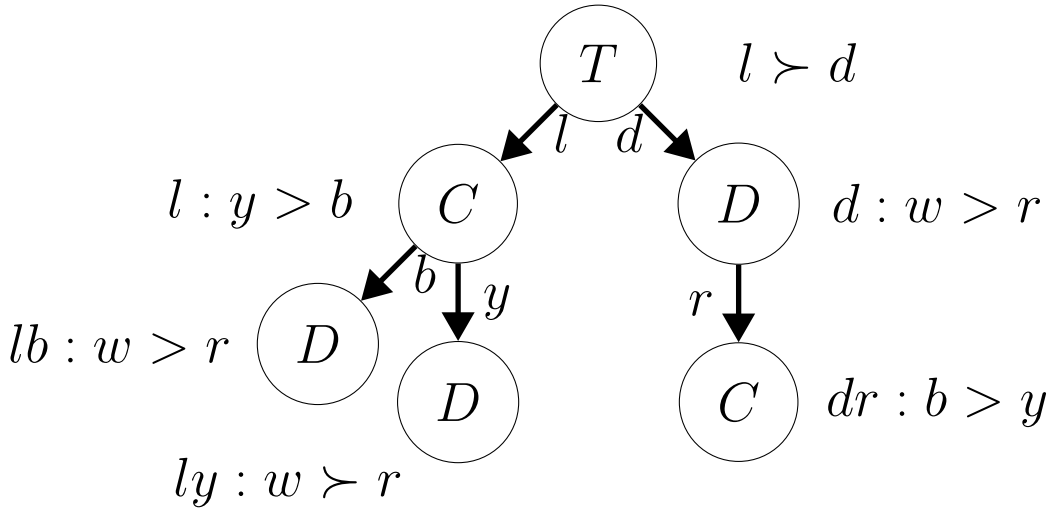
"For desktops, Dvd-unit (read/write) more important than color"

"For laptops, color is more important than the type of Dvd unit"



Conditional local pref. & Conditional variable importance

"For desktops, Dvd-unit (read/write) more important than color"
"For laptops, color is more important than the type of Dvd unit"



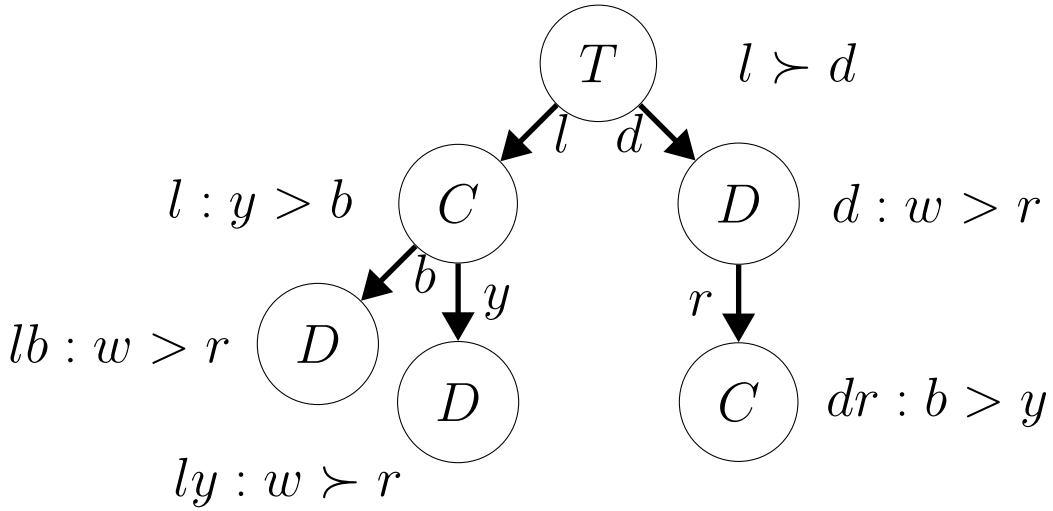
$\Rightarrow$ variable importance tree
$+$ conditional local preference tables

Note : tree need not be complete (but then partial ordering)

Conditional local pref. & Conditional variable importance

"For desktops, Dvd-unit (read/write) more important than color"

"For laptops, color is more important than the type of Dvd unit"



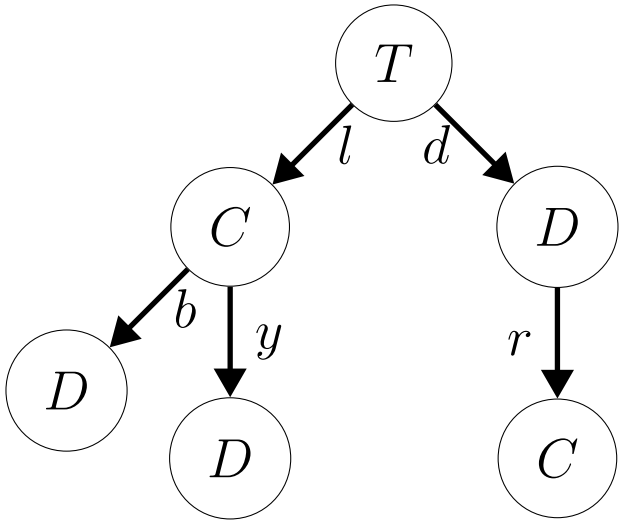
Sample complexity: VC dim $= 2^n - 1$

Active learning: $2^n - 1 + \sum_{k=0}^{n-1} 2^k \log(n-k)$ queries needed

Passive learning: in P (Greedy algorithm still works)

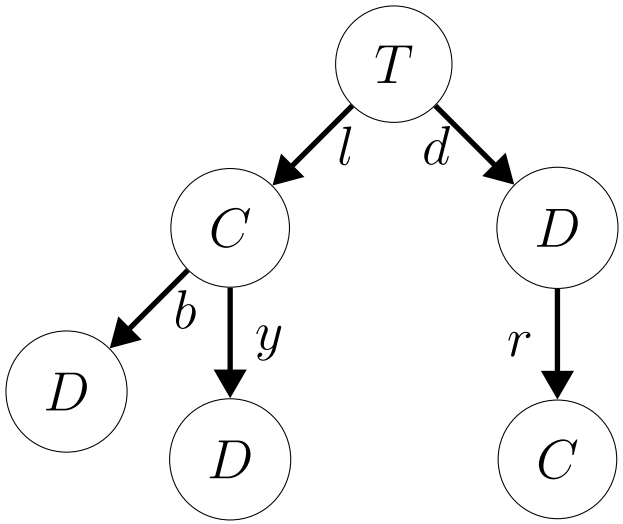
Model optimization: NP-complete

Unconditional local pref. / Conditional variable importance



$y \succ b$
$l \succ d$
$w \succ r$

Unconditional local pref. / Conditional variable importance

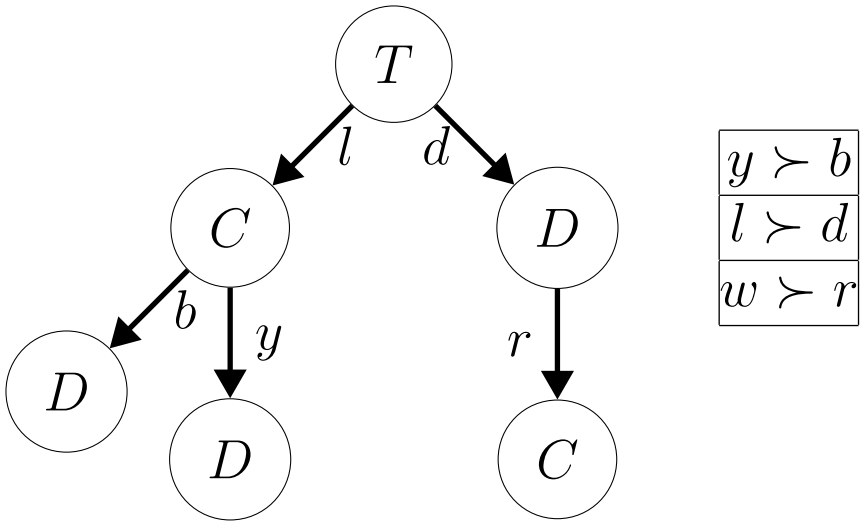


$y \succ b$
$l \succ d$
$w \succ r$

$\Rightarrow$ variable importance tree

$+$ unconditional local preference table

Unconditional local pref. / Conditional variable importance



Sample complexity: ?

Active learning: .

$n + \sum_{k=0}^{n-1} 2^k \log(n-k)$ queries needed (unknown pref.)

$\sum_{k=0}^{n-1} 2^k \log(n-k)$ queries needed (fixed pref.)

Passive learning: NP-complete !! (Greedy algorithm still works)

Model optimization: NP-complete

Quick recap

	VC-dim	active l.	passive l.	approx
UI - FLP		$log(!n)$	P	NP-C
UI - ULP	n	$n + log(!n)$	P	NP-C
UI - CLP	$2^n - 1$	$2^n - 1 + log(!n)$	P	NP-hard
CI - FLP		$g(n)$	P	NP-C
CI - ULP	$\geq n$	$n + g(n)$	NP-C	NP-C
CI - CLP	$2^n - 1$	$2^n - 1 + g(n)$	P	NP-C

$$g(n) = \sum_{k=0}^{n-1} 2^k \log(n-k)$$

Related and future work

- Conditional lexic. orderings introduced by [Wilson, ECAI'06]
  $\Rightarrow$ approximate CP-nets
- Need to explore heuristics to choose variables during execution of the greedy algorithm
- Problem if tree not complete : the ordering is only partial
  $\Rightarrow$ Need to explore mixtures of conditional / unconditional structures
- Need to test algorithms on real / generated data
  $\Rightarrow$ How to deal with noisy date ?