Modelling and Reasoning about Preferences using CP-nets

Source for Slides:
UAI 2014 Tutorial
by Ronen I. Brafman
Ben-Gurion University
Preferences Are Important...

- Important on their own:
  - Needed even when no uncertainty present

- Hard to get:
  - Preferences at least as hard to elicit as likelihood

- Major bottleneck:
  - Major obstacle to the deployment of decision-support and decision automation software
  - Much harder to learn (arguably, impossible)
Preferences Neglected

• In the past 20 years, research focus was on likelihood elicitation
  – Bayesian nets the primary impetus
  – Much work on other formalisms
  – Important applications – primarily diagnosis

• Until recently, little progress in preference representation and management
CP-Nets

An attempt to import the essential ideas behind Bayes Nets into preference modeling

- **Structure:**
  - DiGraph with state variables as nodes
  - Edges denote direct influences

- **Independence**
  - Preferential independence is used to reduce required information
Preferential Independence

If my preferences over the values of a variable \( v \) do not depend on the values of some other variables, then \( v \) is preferentially independent of all other variables.

For processor speed, I prefer 1000 MHz to 800 MHz (all else being equal)

A subset of variables \( X \) is preferentially independent of its complement \( Y = V - X \) if and only if, for all assignments \( x_1, x_2, y_1, y_2 \) the following holds

\[
\bar{x}_1 \bar{y}_1 \geq \bar{x}_2 \bar{y}_1 \quad \text{iff} \quad \bar{x}_1 \bar{y}_2 \geq \bar{x}_2 \bar{y}_2
\]
Conditional Preferential Independence

If my preferences over values of $v$ depend on, and only on, the values of $v_1, \ldots, v_k$, then $v$ is conditionally preferentially independent of $V-\{v_1, \ldots, v_k\}$, given an assignment to $v_1, \ldots, v_k$.

I prefer a 19” screen to a 17” screen if video card is Sony’s

Let $X, Y, Z$ be a partition of $V$ into three disjoint non-empty sets. $X$ is conditionally preferentially independent of $Y$ given $Z$ if and only if, for all $x_1, x_2, y_1, y_2$ the following holds

$$x_1 y_1 z \geq x_2 y_1 z \text{ iff } x_1 y_2 z \geq x_2 y_2 z$$
CP-Nets

• As in BNs, we have
  – Qualitative graphical structure
  – Quantification of the relation between parents & child
    • Quantitative quantification ➔ utility functions
    • Qualitative quantification ➔ preference relations

• Much work concentrates on the qualitative version and assumes no uncertainty

• Important potential applications:
  – Configuration problems
  – Database search
May be I should buy a new car...

Huge assortment of models

Customizable accessories

Customizable mechanics
Preference specification

Optimal automobile

Search ...

Manufacturer Constraints
On a red sport car I prefer a sunroof ...
Applications

- **Product Configuration**
  - Find an optimal feasible configuration

- **Searching large databases on the web**
  - Find best available flight

- **Personalization**
  - Display content most appropriate for user
  - Adapt presentation to user device, preferences
Common Properties

• Uncertainty not a (primary) serious issue
  ➔ Utility functions are not needed
• Lay users
  ➔ No/little training required
  ➔ As effort-less as possible
• On-line/consumer application
  ➔ Expert decision analyst not available
  ➔ Fast response time desirable
What We Want from Preference Model

- Supports simple elicitation process based on *intuitive* and *natural* statements about preferences

On a red sports car I prefer a sunroof ...

- Supports an efficient optimization process
Some Natural Preference Statements

• I prefer 1000 MHz processor to 800 MHz processor

• I prefer 19in screen to 17in screen if the video card is Sony’s

• CPU speed is more important than CPU manufacturer
The Language

• Value preferences
  – Absolute: I prefer $v_1$ to $v_2$ for variables $X$.
  – Conditional: I prefer $v_1$ to $v_2$ for variables $X$ if $Y=y$ and $Z=z$.

• Relative importance
  – Absolute: $X$ is more important than $Y$
  – Conditional: $X$ is more important than $Y$ if $Z=z$
**Interpretation:**

*Ceteris Paribus (CP) Semantics*

*Ceteris Paribus* (Lat.) – All else being equal.

– The preference holds only when comparing two outcomes that differ in the variables mentioned.

• **Example:** “I prefer wine to beer with my meal”

• Interpretation: Given two *identical* meals, one with wine and one with beer, I prefer the former.
Conditional CP Statements

“I prefer red wine to white wine with my meal, ceteris paribus, given that meat is served”

That is: given two identical meals in which meat is served, I prefer red wine to white wine.

Tells us nothing about two identical meals in which meat is NOT served.
CP Statements and Independence

*Ceteris Paribus* preference statements induce independence relations:

- If my preference for wine depends on (and only on) the main course, then wine choice is conditionally preferentially independent of all other variables given the main course value.
CP-nets (Boutilier, Brafman, Hoos, Poole, UAI ‘99)

A qualitative, graphical model of preferences, that captures and organizes statements of conditional preferential independence.

- Each node represents a domain variable.
- The immediate parents $Parents(v)$ of a variable $v$ in the network are those variables that affect user’s preference over the values of $v$.

$Parents(\text{screen size}) = \{ \text{video card manuf.} \}$

$Parents(\text{operating system}) = \{ \text{processor speed, screen size} \}$

Formally: a child is conditionally preferentially independent of all nodes given its parents’ values
CP-nets (Boutilier, Brafman, Hoos, Poole, UAI ‘99)

A qualitative, graphical model of preferences, that captures and organizes statements of conditional preferential independence.

• Each node represents a domain variable.

• The immediate parents $Parents(v)$ of a variable $v$ in the network are those variables that affect user’s preference over the values of $v$.

• A conditional preference table (CPT) is associated with every node in the CP-net
  - Provides an ordering over the values of the node for every possible parent context
Example of a CP-net

\[ a \succ \overline{a} \]

\[ b \succ \overline{b} \]

\[ (a \land b) \lor (\overline{a} \land \overline{b}) : c \succ \overline{c} \]

\[ (a \land \overline{b}) \lor (\overline{a} \land b) : \overline{c} \succ c \]

\[ c : d \succ \overline{d} \]

\[ \overline{c} : d \succ \overline{d} \]

\[ d : f \succ \overline{f} \]

\[ \overline{d} : f \succ \overline{f} \]

\[ c : e \succ \overline{e} \]

\[ \overline{c} : e \succ e \]
CP-nets

• Can be used as a device for helping users describe and structure their preferences
• Can be used as a representation tool for natural language statements
Any **acyclic** CP-net defines a (consistent) partial order over the outcome space.

\[(a \land b) \lor (\bar{a} \land \bar{b}) : c \succ \bar{c}\]

\[(a \land \bar{b}) \lor (\bar{a} \land b) : \bar{c} \succ c\]
Uniqueness

Two fully specified CP-nets are different IFF they induce different partial orders
Example

Dinner Configuration
Suppose that dinner consists of a main course, a soup, and a wine.

Preferences:

• I strictly prefer a steak to a fish fillet.

• I prefer to open with a fish soup if the main course is a steak, and with a vegetable soup if the main course is a fish fillet.

• I prefer a red wine with a vegetable soup, and a white wine with a fish soup.
$s > ff$

$s : fs > vs$
$ff : vs > fs$

$fs : w > r$
$vs : r > w$

Main Course

Soup

Wine

$ff \wedge vs \wedge w$

$s \wedge fs \wedge r$

$s \wedge f s \wedge w$

$s \wedge fs \wedge w$

$s \wedge vs \wedge w$

$s \wedge vs \wedge w$

$s \wedge vs \wedge w$

$s \wedge vs \wedge w$
Relative Importance Relations

• Relative importance statements are very natural
• They express the fact that one variable’s value is more important than another’s
• CP-nets induce *implicit* importance relations between nodes and their descendents
Induced Importance Relations in CP-nets

- fish ≻ vegetable
  - Soup
  - Wine
- child preferences violated
- parent preferences violated

- \( fish \land red \)
- \( vegetable \land red \)
- \( vegetable \land white \)
- \( fish \land white \)
- \( fish : white \gg red \)
- \( vegetable : red \gg white \)
Relative Importance

Processor type is more important to me than operating system (all else being equal).

If it is more important to me that the value of $X$ be high than the value of $Y$ be high, then $X$ is more important than $Y$.

$$X > Y$$

Operating system is more important than processor type (all else being equal), if the PC is used primarily for graphical applications.

If, given $z \in \text{Dom}(Z)$, it is more important to me that the value of $X$ be high than the value of $Y$ be high, then $X$ is conditionally more important than $Y$.

$$X >_z Y$$
\[(a \land b) \lor (\overline{a} \land \overline{b}) : c \succ \overline{c} \]
\[(a \land \overline{b}) \lor (\overline{a} \land b) : \overline{c} \succ c\]

\(a \succ \overline{a}\)

\(b \succ \overline{b}\)
\[ a \succ a \]

\[ a : b \succ \overline{b} \]

\[ \overline{a} : \overline{b} \succ b \]

\[ b : c \succ \overline{c} \]

\[ \overline{b} : \overline{c} \succ c \]

\[ b : d \succ \overline{d} \]

\[ \overline{b} : \overline{d} \succ d \]

\[ S(C, D) = \{ B, E \} \]

\[ be : C \rhd D \]

\[ \overline{be} : D \rhd C \]

\[ b\overline{e} : D \rhd C \]

\[ e \succ \overline{e} \]

- nodes \( \equiv \) variables
- cp-arcs (directed)
- i-arcs (directed)
- ci-arcs (undirected)
- cp-tables
- ci-tables
Example

Choosing a Flight to a Conference in USA
Parameters & Values

• Day of the flight
  – *One* or *Two days* before the conference.

• Airline
  – *British Airlines* or *KLM*.

• Departure time
  – *Morning* or *night*.

• Stop-overs
  – *Direct* flight, or a flight with a *stop-over* in Europe.

• Class
  – *Economy* or *business*. 
My Preferences
Flight Day - $D$

I have a family and much work, so I prefer to leave a day before the conference.

$1d \geq 2d$
Airline - A

I prefer British Airways to KLM because they have a better frequent-flyer program

\[ ba > klm \]
Among the flights leaving two days before the conference I prefer to take an evening/night flight, because it will allow me to work longer on the day of the flight.

However, among the flights leaving one day before the conference I prefer to take a morning/noon flight, because I hate to arrive at the last moment.

\[ 1d : m \succ n \]
\[ 2d : n \succ m \]
Stop-over - S

I am a smoker, and I find long non-smoking day flights difficult to cope with. Thus, I prefer a stop-over in Europe.

However, on night flights I usually sleep (and don’t smoke), thus I prefer direct flights which are shorter.

\[ m : 1l \succ 0l \]

\[ n : 0l \succ 1l \]
Class - C

I sleep well in night flights, regardless of the class, and so at night, I prefer economy which is much cheaper.

During the day I prefer to pay for a seat in business class so that I can enjoy the food, wine, and comfortable seats.

\[
m : b \succ e
\]

\[
n : e \succ b
\]
Day of the flight

Departure Time

Stop-overs

Class

Airline

1d ⊳ 2d

1d : m ⊳ n
2d : n ⊳ m

m : 1l ⊳ 0l
n : 0l ⊳ 1l

m : b ⊳ e
n : e ⊳ b

ba ⊳ klm
Relative Importance

Getting a more preferred flying time is more important to me than getting the preferred airline.

\[ T > A \]
Conditional Relative Importance

1. On a *KLM, day flight*, an intermediate stop in Amsterdam is more important to me than sitting in business class.

2. Given a *British Airways, night flight*, having a direct flight is more important to me than getting a cheaper economy seat.

3. On a *British Airways, day flight*, sitting in business class is more important to me than having a stop-over.

\[ S \succ_{(m \land klm)} C \quad S \succ_{(n \land ba)} C \quad C \succ_{(m \land ba)} S \]
Day of the flight

Departure Time → Airline

Stop-overs ↔ Class

1d ≽ 2d

1d : m ≽ n
2d : n ≽ m

m : 1l ≽ 0l
n : 0l ≽ 1l

m ∩ klm : L ▷ S
n ∩ ba : L ▷ S
m ∩ ba : S ▷ L

m : b ≽ e
n : e ≽ b
Importance: Alternative Specification

“A is more important than B”

“I prefer better values for A regardless of B”.

More generally [Wilson04]:

If Cond then $a_1 > a_2$ regardless of $B_1, \ldots, B_k$. 
Queries on CP-Nets
Queries

• Comparison of two outcomes $o,o'$ given a CP-net $N$:
  – Dominance: does $o \succ o'$ hold according to $N$?
  – Weak Dominance/order: does $o \succcurlyeq o'$ hold according to $N$?

• Set Ordering:
  – ORD: Given a CP-net $N$, order a set of outcomes $O$ consistently with $N$.

• Optimization:
  – Unconstrained: find the optimal outcome
  – Constrained (BEST): given an (explicit/implicit) set of outcomes $O$, find one/some/all elements in $O$ that are not dominated by any other outcome in $O$. 
Dominance queries

Given a CP-net $N$ and a pair of assignments $\alpha$ and $\beta$, determine whether

$$N \models \alpha > \beta$$

If this relation holds, $\alpha$ is preferred to $\beta$, and we say that $\alpha$ dominates $\beta$ with respect to $N$.

A sequence of improving flips from one assignment to another (flipping sequence) is a proof that the latter assignment is preferred to the former.
Dominance testing for CP-nets with binary variables.

<table>
<thead>
<tr>
<th>CP-net graph</th>
<th>Complexity</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree</td>
<td>Linear?</td>
<td>Backtrack free algorithm in BBHP ’99</td>
</tr>
</tbody>
</table>
Dominance testing for CP-nets with binary variables.

<table>
<thead>
<tr>
<th>CP-net graph</th>
<th>Complexity</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree</td>
<td>$O(n^2)$</td>
<td>Lower bound</td>
</tr>
</tbody>
</table>

Diagram of a tree CP-net graph.
Dominance testing for CP-nets with binary variables.

<table>
<thead>
<tr>
<th>CP-net graph</th>
<th>Complexity</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree</td>
<td>$O(n^2)$</td>
<td>Lower bound</td>
</tr>
<tr>
<td>Polytrees</td>
<td>$O(2^{2k}n^{2k+3})$</td>
<td>$k$ – maximal indegree</td>
</tr>
</tbody>
</table>
Dominance testing for CP-nets with binary variables.

<table>
<thead>
<tr>
<th>CP-net graph</th>
<th>Complexity</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree</td>
<td>$O(n^2)$</td>
<td>Lower bound</td>
</tr>
<tr>
<td>Polytree</td>
<td>$O(2^{2k} n^{2k+3})$</td>
<td>$k$ - maximal indegree</td>
</tr>
<tr>
<td>Singly-connected DAG</td>
<td>NP-complete</td>
<td>Reduction from 3SAT</td>
</tr>
</tbody>
</table>
Dominance testing for CP-nets with binary variables.

<table>
<thead>
<tr>
<th>CP-net graph</th>
<th>Complexity</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree</td>
<td>$O(n^2)$</td>
<td>Lower bound</td>
</tr>
<tr>
<td>Polytree</td>
<td>$O(2^{2k} n^{2k+3})$</td>
<td>$k$ - maximal indegree</td>
</tr>
<tr>
<td>Singly-connected DAG</td>
<td>NP-complete</td>
<td>Reduction from 3SAT</td>
</tr>
<tr>
<td>$\delta$-connected DAG</td>
<td>NP-complete</td>
<td>Minimal flipping sequences are polynomially bounded</td>
</tr>
</tbody>
</table>
Dominance testing for CP-nets with binary variables.

<table>
<thead>
<tr>
<th>CP-net graph</th>
<th>Complexity</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree</td>
<td>$O(n^2)$</td>
<td>Lower bound</td>
</tr>
<tr>
<td>Polytree</td>
<td>$O(2^{2k}n^{2k+3})$</td>
<td>$k$ - maximal indegree</td>
</tr>
<tr>
<td>Singly-connected DAG</td>
<td>NP-complete</td>
<td>Reduction from 3SAT</td>
</tr>
<tr>
<td>$\delta$-connected DAG</td>
<td>NP-complete</td>
<td>Minimal flipping sequences are polynomially bounded</td>
</tr>
<tr>
<td>DAG</td>
<td>?</td>
<td>NP or harder?</td>
</tr>
</tbody>
</table>
Pair Ordering – A Cheaper Alternative

Dominance query:

Given a CP-net $N$ and a pair of assignments $\alpha$ and $\beta$, ask whether $N \models \alpha > \beta$.

Ordering query:

Given a CP-net $N$ and a pair of assignments $\alpha$ and $\beta$, ask whether $N \not\models \beta > \alpha$.

If $N \not\models \beta > \alpha$, there exists a complete (total) preference ordering consistent with $N$ in which $\alpha > \beta$.

In such a case we say that $\alpha$ is consistently orderable over $\beta$ with respect to $N$. 
Claim 1:

Let $N$ be a CP-net, and $\alpha$, $\beta$ be a pair of complete assignments. If there exists a variable $X$ in $N$, such that:

1. $\alpha$ and $\beta$ assign the same values to all ancestors of $X$ in $N$, and

2. given the assignment provided by $\alpha$ (and $\beta$) to $\text{Parents}(X)$, $\alpha$ assigns a more preferred value to $X$ than that assigned by $\beta$

then $N \not\models \beta > \alpha$. 
Condition provided by Claim 1 is:

😊 Testable in time linear in the number of variables,

😢 Sufficient BUT not necessary for $N \not\models \beta > \alpha$.

“Partial necessity” – either $N \not\models \beta > \alpha$ or $N \not\models \alpha > \beta$ (or both) will be determined by the procedure used to answer the ordering queries.
Claim 2:

Given a CP-net $N$ over $n$ variables and a set of complete assignments $o_1, \ldots, o_m$, ordering these assignments consistently with $N$ can be done using ordering queries only, in time $O(nm^2)$. 
Optimization
Finding the preferentially optimal outcome for an acyclic network is straightforward!

\[ a \succ \overline{a} \]

\[ b \succ \overline{b} \]

\[ (a \land b) \lor (\overline{a} \land \overline{b}) : c \succ \overline{c} \]

\[ (a \land \overline{b}) \lor (\overline{a} \land b) : \overline{c} \succ c \]

\[ c : d \succ \overline{d} \]

\[ \overline{c} : \overline{d} \succ d \]

\[ d : f \succ \overline{f} \]

\[ \overline{d} : \overline{f} \succ f \]
Constrained Optimization

Input:
• Constraints (defining what’s feasible)
• Preferences (defining what’s desirable)

Output:
• One undominated, feasible solution or
• A set of undominated, feasible solutions or
• All undominated, feasible solutions
Solving Constrained Optimization Problems

• Basic idea: Generate & Test
  – Generate outcomes
  – Test for feasibility
  – Test for optimality

• Looks bad – testing for optimality is difficult
Ordered Generate & Test

• Generate outcomes in non-increasing order
• Test for feasibility
• Check for optimality:
  – First feasible outcome is optimal!
  – If more than one is needed:
    • Maintain set of optimal solutions
    • Compare new feasible solutions against current optimal set using dominance testing
Summary

• Preferences are important!
• We need them to support user decisions
• We need to develop tools for eliciting, representing, and reasoning with them
• CP-nets attempt to utilize ideas that have been successful in probabilistic reasoning -- **structure and independence** -- to provide such capabilities