Aide à la décision/ Decision aid

Souhila KACI

Partie 1/Part 1
Représentation des informations avec priorités/Representation of prioritized information

Modeling

- Représentation compacte
  - modélisation logique
    - pondérée
      - positivité
      - négativité
    - conditionnelle
  - modélisation graphique

- The outcomes (objects, products, states of the world, etc) that need to be compared are usually of a combinatorial nature, i.e. defined by the values they assign to a set of variables
- $V = \{X_1, \cdots, X_n\}$: a set of variables
- $v(X_i) \in Dom(X_i)$
- $\Pi_{i=1}^{n} Dom(X_i)$: the set of possible outcomes

  ↳ produit cartésien des variables
- $\Omega \subseteq \Pi_{i=1}^{n} Dom(X_i)$: the set of feasible outcomes

  déterminé par des contraintes d'intégrité
- ω: an outcome

Example

- $V = \{V_1, V_2, V_3\}$, V_1, V_2 and V_3 respectively standing for "dish", "wine" and "dessert"
- $Dom(V_1) = \{fish, meat\}$, $Dom(V_2) = \{red, white, rosé\}$, $Dom(V_3) = \{cake, ice\_cream\}$
- $\Omega = \{\omega_0 = fish - red - cake, \omega_1 = fish - red - ice\_cream,$ $\omega_2 = fish - white - cake, \omega_3 = fish - white - ice\_cream,$ $\omega_4 = fish - rosé - cake, \omega_5 = fish - rosé - ice\_cream,$ $\omega_6 = meat - red - cake, \omega_7 = meat - red - ice\_cream,$ $\omega_8 = meat - white - cake, \omega_9 = meat - white - ice\_cream,$ $\omega_{10} = meat - rosé - cake, \omega_{11} = meat - rosé - ice\_cream\}$

Modeling

- Outcomes have varied uncertainty/priority/satisfaction. We speak about an uncertainty/priority/satisfaction relation. We use the generic term preference/plausibility relation
- Modeling is the mathematical writing of a preference/plausibility relation
- We distinguish between ordinal representation and cardinal representation

- An ordering relation (generally called preference relation): $\succeq$, $\succ$
- ω is at least as preferred (plausible) as ω' ($\omega \succeq \omega'$)
- ω is strictly preferred to (more plausible than) ω' ($\omega \succ \omega'$)
- ω and ω' are equally preferred (plausible) ($\omega \approx \omega'$)
- ω and ω' are incomparable ($\omega \sim \omega'$)

- $\succeq$ is a preorder over Ω iff $\succeq$ is
  - reflexive: $\forall \omega \in \Omega$, $\omega \succeq \omega$
  - transitive: $\forall \omega, \omega', \omega'' \in \Omega$, if $\omega \succeq \omega'$ and $\omega' \succeq \omega''$ then $\omega \succeq \omega''$
- $\succ$ is an order over Ω iff $\succeq$ is
  - irreflexive: $\forall \omega \in \Omega$, $\omega \succ \omega$ is not true ($not(\omega \succ \omega)$)
  - transitive: $\forall \omega, \omega', \omega'' \in \Omega$, if $\omega \succ \omega'$ and $\omega' \succ \omega''$ then $\omega \succ \omega''$
- $\succeq$ (resp. $\succ$) is complete iff all outcomes are comparable. Otherwise it is partial.
- If $\succeq$ is a complete preorder then it can be written under a well ordered partition of the form $(E_1, \cdots, E_n)$ such that $\forall \omega, \omega' \in \Omega$, $\omega \succeq \omega'$ iff $\omega \in E_i$, $\omega' \in E_j$ with $i \leq j$.
  This also applies to a complete order. Each stratum is then composed of one outcome only.

Examples

$\Omega = \{\omega_0 = fish - red, \omega_1 = fish - white,\ \omega_2 = fish - rosé,$
$\omega_3 = meat - red, \omega_4 = meat - white, \omega_5 = meat - rosé\}$

- $\succeq_1$: $fish - white \approx meat - red \succ fish - red \approx meat - white \succ fish - rosé \approx meat - rosé$ is a complete preorder
- $\succeq_2$: $fish - white \approx meat - red \succ fish - red \approx meat - white,$ $fish - rosé \approx meat - rosé$ is a partial preorder
- $\succ_3$: $fish - white \succ meat - red \succ fish - red \succ fish - rosé \succ meat - white \succ meat - rosé$ is a complete order
- $\succ_4$: $fish - white \succ meat - red \succ fish - red \succ fish - rosé,$ $meat - white \succ meat - rosé$ is a partial order
- $\succeq_1 = (\{fish - white, meat - red\},$ $\{fish - red, meat - white\}, \{fish - rosé, meat - rosé\})$
- $\succ_3 = (\{fish - white\}, \{meat - red\}, \{fish - red\},$ $\{fish - rosé\}, \{meat - white\}, \{meat - rosé\})$

Modeling: Cardinal representation

- A numerical function u which associates with each outcome ω a numerical value $u(\omega)$
- ω is strictly preferred to (more plausible than) ω' iff $u(\omega) > u(\omega')$
- ω and ω' are equally preferred (plausible) iff $u(\omega) = u(\omega')$
- Incomparability cannot be expressed

Examples

- $u_1(fish - white) = u_1(meat - red) = 10,$
  $u_1(fish - red) = u_1(meat - white) = 8,$
  $u_1(fish - rosé) = u_1(meat - rosé) = 3$

- $u_2(fish - white) = 25,\ u_2(meat - red) = 20,$
  $u_2(fish - red) = 18,\ u_2(fish - rosé) = 15,$
  $u_2(meat - white) = 9,\ u_2(meat - rosé) = 7$

What is your preference?

When Maria was asked which juice she would prefer, she immediately said: orange juice.

When she was doing her shopping, she hesitated between red skirt and white pants but she finally choose the former since she prefers red to white and skirt to pants.

But when she was asked to choose the composition of a meal based on main dish (fish or meat), wine (red, white or rosé) and dessert (cake or ice cream), the choice was less obvious! She said "I prefer fish to meat", "if fish is served then I prefer white wine otherwise I prefer red wine" and "if cake is served then I prefer meat otherwise I prefer fish".

Maria much more hesitated when she had to choose among three professor positions!

Unfortunately things are not too simple!

The elicitation of a preference relation is a hard task

What's your preference among the menus *fish – red – cake* and *meat – white – ice_cream*?

- I prefer *fish – red – cake* to *meat – white – ice_cream*
- I prefer *meat – white – ice_cream* to *fish – red – cake*
- I have the same preference for both
- They are incomparable
- I don't know, not easy to make a choice! But...
  - Partial preferences
  - I prefer fish to meat, if fish is served then I prefer white wine otherwise I prefer red wine, I really like fish, I like fish with weight .9 and meat with weight .6, etc
  - We need representation languages to support such preferences

Knowledge representation languages

A preference/plausibility relation over Ω (called a model) is associated with each language

Knowledge representation languages fall into two categories

- Logical languages: weighted logics, conditional logics
- Graphical languages

Knowledge representation languages: Weighted logics

What is a weighted logic?

A weighted logic associates certainty/priority degrees with propositional logic formulas. It may be qualitative or quantitative.

Outline

- Possibilistic logic
- Penalty logic

- It encodes a numerical function, called a possibility distribution π from Ω to $[0, 1]$. $\pi(\omega)$ is the plausibility/satisfaction associated with ω
- $\pi(\omega) = 1$: nothing prevents ω from being plausible/ satisfactory → peut encore baisser
- → définitivement non
  $\pi(\omega) = 0$: ω is certainly not plausible/satisfactory
- $\pi(\omega) > \pi(\omega')$ iff ω is more plausible/satisfactory than ω'

- A general knowledge base $\Sigma = \{(\phi_i, a_i) | i = 1, \ldots, n\}$
- The associated possibility distribution should satisfy the following constraints:

$$\forall i = 1, \ldots, n \; \Pi(\neg \phi_i) \leq 1 - a_i$$

- The unique possibility distribution associated with Σ is computed in the following way: $\forall \omega \in \Omega$,

$$\pi(\omega) = \begin{cases} 1 \quad \text{↳ satisfait toutes les connaissances} & \text{if } \omega \models \phi_1 \wedge \cdots \wedge \phi_n \\ 1 - \max\{a_i | (\phi_i, a_i) \in \Sigma, \omega \not\models \phi_i\} & \text{otherwise.} \end{cases}$$

↳ sinon 1- la valeur associée à la connaissance la plus importante

Example

Example 1

Let
$\Sigma = \{(p, 1), (\neg p \vee b, .8), (\neg p \vee \neg f, .8), (\neg b \vee f, .4), (\neg b \vee w, .4)\}$

Example 2

Let $\Sigma = \{(\neg p \vee b, .8), (\neg p \vee \neg f, .8), (\neg b \vee f, .4), (\neg b \vee w, .4)\}$

A penalty base is a set of weighted formulas of the form

$$\Sigma = \{(\phi_i, a_i) | i = 1, \ldots, n\}$$

with a_i is a real number.
The associated penalty distribution is computed as follows: $\forall \omega \in \Omega$,

$$p(\omega) = \begin{cases} 0 \text{ si satisfait toutes les connaissances} & \text{if } \omega \models \phi_1 \wedge \cdots \wedge \phi_n \\ \sum \{a_i | (\phi_i, a_i) \in \Sigma, \omega \not\models \phi_i\} & \text{otherwise.} \end{cases}$$

↳ sinon, somme des valeurs associées aux formules non-satisfaites

$$p(\omega) < p(\omega') \text{ iff } \omega \text{ is more plausible/satisfactory than } \omega'$$

Penalty logic (2)

Example

Let
$\Sigma = \{(p, 100), (\neg p \vee b, 80), (\neg p \vee \neg f, 80), (\neg b \vee f, 40), (\neg b \vee w, 40)\}$

Knowledge representation languages: Conditional logics

Basic ingredients

- (Conditional) comparative preference statements
  - Prefer A to B
  - If C is true, prefer A to B (equivalent to prefer $C \wedge A$ to $C \wedge B$)
- Both statements can be written as $p \rhd q$ (prefer p to q)
- Comparative preference statements offer a simple and intuitive way for expressing preferences
- However they also come with difficulties regarding their interpretation

Problem 1: Common outcomes

Example

- Let V_1 and V_2 respectively stand for "dish" and "wine" with $Dom(V_1) = \{fish, meat\}$ and $Dom(V_2) = \{red, white\}$.
- Let $fish \rhd white$.
- We have to compare $\{fish - red, fish - white\}$ and $\{meat - white, fish - white\}$
- $fish - white$ belongs to both sets!

"$p \wedge \neg q$ is preferred to $q \wedge \neg p$"(von Wright principle)

$\hookrightarrow p \rhd q \equiv (p \wedge \neg q) \wedge \neg(\neg p \wedge q) \rhd \neg(p \wedge \neg q) \wedge (\neg p \wedge q)$

Example

$fish \rhd white$ stands for $fish \wedge \neg white \rhd white \wedge \neg fish$ (i.e., $fish - red \rhd meat - white$)

Problem 2: Comparison of two sets of objects

How do we compare $p \wedge \neg q$-outcomes and $q \wedge \neg p$-outcomes ?

How should we interpret "prefer (p=fish) to (q=meat)"?

- strong semantics: p is always preferred to q ($\forall \rhd \forall$)
  any $p \wedge \neg q$-outcome is preferred to any $q \wedge \neg p$-outcome
- ceteris paribus semantics: ($px \succ qx$, $px \not\succ qy$)
  any $p \wedge \neg q$-outcome is preferred to any $q \wedge \neg p$-outcome, if the two outcomes are completed in the same way
- optimistic semantics: at least one $p \wedge \neg q$-outcome is preferred to any $q \wedge \neg p$-outcome ($\exists \rhd \forall$)
- pessimistic semantics:
  at least one $q \wedge \neg p$-outcome is less preferred to any $p \wedge \neg q$-outcome ($\forall \rhd \exists$)
- opportunistic semantics: ($\exists \rhd \exists$)
  at least one $p \wedge \neg q$-outcome is preferred to at least one $q \wedge \neg p$-outcome

Additional definitions

Definition: Best/Worst outcomes w.r.t. $\succeq$

- $\max(\Omega, \succeq) = \{\omega | \omega \in \Omega, \nexists \omega' \in \Omega, \omega' \succ \omega\}$
- $\min(\Omega, \succeq) = \{\omega | \omega \in \Omega, \nexists \omega' \in \Omega, \omega \succ \omega'\}$
- $\max(p, \succeq) = \{\omega | \omega \in Mod(p), \nexists \omega' \in Mod(p), \omega' \succ \omega\}$
- $\min(p, \succeq) = \{\omega | \omega \in Mod(p), \nexists \omega' \in Mod(p), \omega \succ \omega'\}$

An equivalent reading of the semantics

- strong semantics:
  The worst ranked $p \wedge \neg q$-outcome is preferred (w.r.t. $\succeq$) to the best ranked $q \wedge \neg p$-outcome

- optimistic semantics:
  The best ranked $p \wedge \neg q$-outcome is preferred (w.r.t. $\succeq$) to the best ranked $q \wedge \neg p$-outcome

- pessimistic semantics:
  The worst ranked $p \wedge \neg q$-outcome is preferred (w.r.t. $\succeq$) to the worst ranked $q \wedge \neg p$-outcome

- opportunistic semantics:
  The best ranked $p \wedge \neg q$-outcome is preferred (w.r.t. $\succeq$) to the worst ranked $q \wedge \neg p$-outcome

A preference set $\mathcal{P}_{\rhd} = \{p_i >_{\rhd} q_i | i = 1, \cdots, n\}$ is a set of preference statements obeying the same semantics, $\rhd = st, cp, opt, pes, opp$ (for strong, ceteris paribus, optimistic, pessimistic, opportunistic)

Example

$$\mathcal{P}_{\rhd} = \{fish \rhd meat,$$
$$red \wedge cake \rhd white \wedge ice\_cream,$$
$$fish \wedge white \rhd fish \wedge red\}$$

- How to rank-order menus (the set Ω) w.r.t. $\mathcal{P}_{\rhd}$?
- $\succeq$ satisfies $\mathcal{P}_{\rhd}$ iff $\succeq$ satisfies each statement $p_i \rhd q_i$ in $\mathcal{P}_{\rhd}$
- Several complete preorders may satisfy the set $\mathcal{P}_{\rhd}$. They are called models of $\mathcal{P}_{\rhd}$.

Example

$$\Omega = \{fish - red, fish - white, meat - red, meat - white\}$$

$$\mathcal{P}_{opt} = \{fish >_{opt} meat\}$$

- $\succeq_1 = (\{fish - red, fish - white\}, \{meat - red, meat - white\})$,
  $\succeq_2 = (\{fish - red\}, \{fish - white, meat - red, meat - white\})$,
  $\succeq_3 = (\{fish - white\}, \{fish - red, meat - red, meat - white\})$,
- $\succeq_4 = (\{fish - red, fish - white\}, \{meat - red\}, \{meat - white\})$,
  $\succeq_5 = (\{fish - red, fish - white\}, \{meat - white\}, \{meat - red\})$,
- ... ↳ Au moins une relation sans préférence → incomparabilité

- $fish - red$? $meat - red$: $fish - red \sim meat - red$
- $fish - red$? $meat - white$: $fish - red \sim meat - white$
- $fish - white$? $meat - red$: $fish - white \sim meat - red$
- $fish - white$? $meat - white$: $fish - white \sim meat - white$

→ il faut choisir une des relations

- Minimal specificity principle: each outcome is put in the highest possible level in the preorder

Principle

An outcome is satisfactory unless there is a reason to state the contrary

Example

- $\mathcal{P}_{opt} = \{fish >_{opt} meat\}$
- $\succeq_1 = (\{fish - red, fish - white\}, \{meat - red, meat - white\})$
- $\succeq_2 = (\{fish - red\}, \{fish - white, meat - red, meat - white\})$
- $\succeq_3 = (\{fish - white\}, \{fish - red, meat - red, meat - white\})$
- $\succeq_4 = (\{fish - red, fish - white\}, \{meat - red\}, \{meat - white\})$
- $\succeq_5 = (\{fish - red, fish - white\}, \{meat - white\}, \{meat - red\})$
- ...

Selecting a unique model: Specificity principle (2)

- Maximal specificity principle: each outcome is put in the lowest possible level in the preorder

Principle

An outcome is not satisfactory unless there is a reason to state the contrary

Example

- $\mathcal{P}_{pes} = \{fish >_{pes} meat\}$
- $\succeq_1 = (\{fish - red, fish - white\}, \{meat - red, meat - white\})$
- $\succeq_2 = (\{fish - red\}, \{fish - white, meat - red, meat - white\})$
- $\succeq_3 = (\{fish - white\}, \{fish - red, meat - red, meat - white\})$
- $\succeq_4 = (\{fish - red, fish - white\}, \{meat - red\}, \{meat - white\})$
- $\succeq_5 = (\{fish - red, fish - white\}, \{meat - white\}, \{meat - red\})$
- ...

Minimal/maximal specificity principles

Uniqueness of the models

- The least specific model of $\mathcal{P}_{opt}$ (resp. $\mathcal{P}_{cp}$, $\mathcal{P}_{st}$) exists.
- The most specific model of $\mathcal{P}_{pes}$ (resp. $\mathcal{P}_{cp}$, $\mathcal{P}_{st}$) exists.
- The most specific model of $\mathcal{P}_{opt}$ (resp. $\mathcal{P}_{opp}$) doesn't exist.
- The least specific model of $\mathcal{P}_{pes}$ (resp. $\mathcal{P}_{opp}$) doesn't exist.

Algorithms to compute the unique models

- Input: $\mathcal{P} = \{s_i : p_i \rhd q_i | i = 1, \cdots, n\}$ (s_i for statement)

- We define $\mathcal{L}(\mathcal{P}) = \{(L(s_i), R(s_i)) | s_i \in \mathcal{P}\}$ with $L(s_i) = Mod(p_i \wedge \neg q_i)$ and $R(s_i) = Mod(q_i \wedge \neg p_i)$

  Mod (F): ensemble des outcomes (ω) qui vérifient F

  (left) (right)

- Output: a unique model (complete preorder) following minimal/maximal specificity principles depending on the semantics

- We focus on **optimistic** semantics

Optimistic semantics

- It is a left-hand weakening of strong semantics. It requires at least one $p \wedge \neg q$-outcomes to be preferred to all $q \wedge \neg p$-outcomes.
- It obeys minimal specificity principle.

Optimistic semantics - Minimal specificity principle

1. $l = 0$
2. While $\Omega \neq \emptyset$

   On met dans la strate courante les éléments qui ne sont à droite dans aucune des contraintes

   - $l = l + 1$
   - $E_l = \{t | t \in \Omega, \nexists (L(s_i), R(s_i)) \in \mathcal{L}(\mathcal{P}_\triangleright), t \in R(s_i)\}$
   - If $E_l = \emptyset$ then stop (inconsistent preferences), $l = l - 1$
   - $\Omega = \Omega \backslash E_l$
   - remove $(L(s_i), R(s_i))$ with $L(s_i) \cap E_l \neq \emptyset$ (remove satisfied preferences) → contraintes contenant à gauche un outcome ajouté à la strate
   - return to 2
3. Output: $\succeq = (E_1, \ldots, E_l)$

Example

Example

Let V_1 and V_2 with $Dom(V_1) = \{fish, meat\}$ and $Dom(V_2) = \{white, red\}$.
Let $\mathcal{P} = \{fish \rhd meat, red \wedge meat \rhd red \wedge fish\}$.

Beyond semantics

Optimistic/Pessimistic semantics

- Optimistic: What is not explicitly rejected is satisfactory
- Pessimistic: What is not explicitly desired is not satisfactory

Example

$fish \rhd meat$, $red : meat \rhd fish$

$\{fish - white, fish - red\} \rhd \{meat - white, meat - red\}$

$\{meat - red\} \rhd \{fish - red\}$

- Strong, ceteris paribus semantics: $fish - red \succ meat - red$ and $meat - red \succ fish - red$
- Optimistic semantics:
  $\succeq = (\{fish - white\}, \{meat - white, meat - red\}, \{fish - red\})$
- Pessimistic semantics:
  $\succeq = (\{meat - red\}, \{fish - white, fish - red\}, \{meat - white\})$

Exercise 1

Suppose an individual is planning a holiday. She expresses her preferences on the basis of three variables: P (for period) which is either W or S (Winter and Summer resp.), D (for destination) which is either M or B (Mountain and Beach resp.) and L (for location) which is either H or A (Hotel and Apartment resp.). The individual expresses three preference statements:

(i) she would prefer travel in winter than in summer,

(ii) if destination is beach then she would prefer travel in summer than in winter,

(iii) if she travels in winter than she would prefer rent an apartment than a hotel.

Exercise 2

$\mathcal{P} = \{p \rightarrow b, p \rightarrow \neg f, b \rightarrow f, b \rightarrow w\}$ (new notation!)

Knowledge representation languages: Graphical representations

- The elicitation of a utility function (or preference relation) is much easier when it exhibits a particular structure.
- We speak about preference independence between variables.
- A set of variables V_1 is independent of the set of variables V_2 if and only if preferences over V_1 can be stated given a fixed value of variables in V_2. $\rightarrow$ $\left.\begin{array}{l} a : b \succ \neg b \\ \neg a : \neg b \succ b \end{array}\right\}$ b dependant of a

Important

Independence is not commutative

If X_i is independent of X_j (X_i and X_j are two variables) then this does not necessarily mean that X_j is independent of X_i.

Example

A user's preference over the main dish may be independent of the wine. Therefore she prefers *fish* to *meat* given a fixed value of wine. We have $fish - white \succ meat - white$ and $fish - red \succ meat - red$. However her preference over wine depends on the main dish. Therefore, she prefers *white wine* with *fish* and *red wine* with *meat*.

Preferential independence

Let $\succeq$ be a preference relation. $X \subseteq V$ is preferentially independent of $Y = V \backslash X$ w.r.t. $\succeq$ if and only if for all $x, x' \in Asst(X), y, y' \in Asst(Y)$, we have

$$xy \succeq x'y \text{ iff } xy' \succeq x'y'.$$

This means that the preference relation over values of X, when all other variables get a fixed value, is the same regardless the values of these variables. This is the qualitative counterpart of the additive independence property of a utility function. We say that x is preferred to x' ceteris paribus.

Conditional preferential independence

Let $\succeq$ be a preference relation. Let X, Y and Z be a partition of V. X and Y are conditionally preferentially independent given $z \in Asst(z)$ w.r.t. $\succeq$ if and only if for all $x, x' \in Asst(X), y, y' \in Asst(Y)$, we have

$$xyz \succeq x'yz \text{ iff } xy'z \succeq x'y'z.$$

This means that X and Y are preferentially independent in the sense of the previous definition (Preferential independence) only when Z is assigned the value z.

Conditional Preference Networks (3)– CP-nets

- CP-nets exploit conditional preferential independence in structuring a user's preferences.
- They are graphical languages which consist of nodes and arrows that connect the nodes.
- Each node represents a variable at hand.

Conditional Preference Networks (4)– CP-nets

A CP-net, let us say N, is constructed as follows:

1. for each variable X_i, the user specifies a set of parent variables, denoted $Pa_N(X_i)$, that affect her preference over the values of X_i. This preferential dependency is represented in the graph by an arrow connecting each node representing a parent variable in $Pa_N(X_i)$ to the node representing X_i. By abuse of language, we simply speak about the node X_i (instead of the node representing X_i) and parent nodes. The set $Pa_N(X_i)$ may be empty which is interpreted as the user specifying her preference over the values of X_i independently of the values of the remaining variables. In this case, X_i is called a root node.

2. The user specifies a preference order over the values of X_i for all instantiations of the variable set $Pa_N(X_i)$. Therefore, the node X_i in the graph is annotated with a conditional preference table $CPT(X_i)$ representing these preferences.

Conditional Preference Networks (5)– CP-nets

- for root nodes X_i, the conditional preference table $CPT(X_i)$ provides the strict preference from among x_i and $\neg x_i$ (suppose that we act over binary variables), other things being equal, i.e., $\forall y \in Asst(Y)$, $x_i y \succ \neg x_i y$, where $Y = V \backslash \{X_i\}$; this is the ceteris paribus semantics; this means that X is preferentially independent of Y; in $CPT(X_i)$ this preference is written $x_i > \neg x_i$;

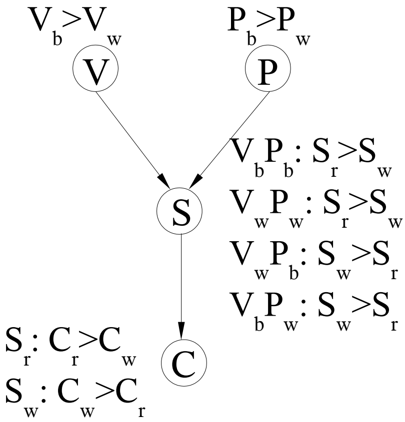
- for other nodes X_j, $CPT(X_j)$ describes the preferences from among x_j and $\neg x_j$, other things being equal, given any assignment of $Pa_N(X_j)$, i.e., $x_j zy \succ \neg x_j zy$, $\forall z \in Asst(Pa_N(X_j))$ and $\forall y \in Asst(Y)$, where $Y = V \backslash (\{X_j\} \cup Pa_N(X_j))$; this means that X is preferentially independent of Y given Z; in the preference table $CPT(X_j)$ we write $z : x_j > \neg x_j$ for each assignment z of $Pa_N(X_j)$.

- Consider four binary variables $V(vest)$, $P(pants)$, $S(shirt)$ and $C(shoes)$ with
  $Dom(V) = \{V_b, V_w\}$, $Dom(P) = \{P_b, P_w\}$,
  $Dom(S) = \{S_r, S_w\}$ and $Dom(C) = \{C_r, C_w\}$.
- Assume that when choosing his evening outfit, Peter is not able to compare the sixteen outcomes but expresses the following preferences over partial descriptions of outcomes:
  - (P_1): he prefers a black vest to a white vest,
  - (P_2): he prefers black pants to white pants,
  - (P_3): when vest and pants have the same color, he prefers a red shirt to a white shirt; otherwise, he prefers a white shirt, and
  - (P_4): when the shirt is red, he prefers red shoes; otherwise, he prefers white shoes.

The problem now is how to rank-order the possible outcomes according to Peter's preferences.

Example (2)

Représentation des dépendances



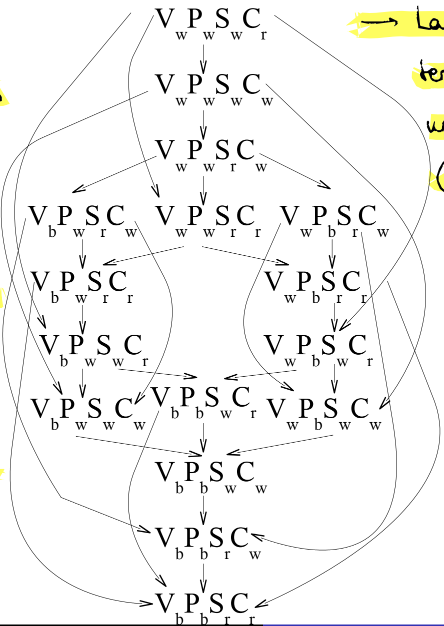
The preference relation over Ω associated with a CP-net N, denoted by $\succeq_N$, is the deductive closure of all local preferences induced by the conditional preference tables of N between completely specified outcomes. Generally, $\succeq_N$ is a partial order and represented by its associated strict preference relation $\succ_N$. When the CP-net is acyclic, its associated preference relation is acyclic too.

Example (3): The associated partial order

Représentation des préférences

¬a ▷ a devient ¬a ←— a (flèche vers préférence)

deux préférences sont comparables s'il existe un chemin



→ La plupart du temps, on obtient un ordre partiel (≠ conditionnelle)

Preference queries

- Due to the ceteris paribus semantics, strict preferences induced by CPT hold among outcomes which differ only in the value of one variable. This is called "worsening flip".
- $\succ_N$ is the deductive closure of local preferences induced by the conditional preference tables.
- Thus, the preferential comparison of two outcomes w.r.t. $\succ_N$ is limited to the pairs for which there exists a path between them through a sequence in which two successive outcomes differ only in the value of one variable.
- For example $V_b P_b S_w C_w$ is preferred to $V_w P_b S_r C_w$ thanks to the following sequence of worsening flips: $V_b P_b S_w C_w \succ_N V_w P_b S_w C_w \succ_N V_w P_b S_w C_r \succ_N V_w P_b S_r C_r \succ_N V_w P_b S_r C_w$.

Important

Not all partial orders can be compactly represented by a CP-net.