Teorie užitku

- Většinou měříme výplatu, hodnotu atd. penězi.
- MEU (maximalizace očekávaného zisku) je většinou rozumná věc k volbě.
- Ale občas je lidská intuice jiná a je na nás, jestli věříme víc intuici nebo tomu, že máme maximalizovat aditivní MEU.

Loterie

Mějme generátor náhodného výsledku podle zadané pravděpodobnosti. Můžete se účastnit jedné ze dvou loterií, otázka zní, kterou si vyberete.
Odpovídejte na základě intuice, pak teprve max. MEU.

Loterie A

1. 80% šance vyhrát \$400
2. 100% šance vyhrát \$300

Co zvolíte?

Loterie B

A z této dvojice?

1. 20% šance vyhrát \$400
2. 25% šance vyhrát \$300

Co zvolíte?

Loterie

Mějme generátor náhodného výsledku podle zadané pravděpodobnosti. Můžete se účastnit jedné ze dvou loterií, otázka zní, kterou si vyberete.
Odpovídejte na základě intuice, pak teprve max. MEU.

Loterie A

1. 80% šance vyhrát \$400
2. 100% šance vyhrát \$300

Co zvolíte?

Loterie B

A z této dvojice?

1. 20% šance vyhrát \$400
2. 25% šance vyhrát \$300

Co zvolíte?

Užitek peněz

Loterie

Opět výběr ze dvou loterií:

- Buď dostat \$1000000
- nebo hodit korunou a na 50% dostat \$3000000

Užitek peněz

- Ekonomové říkají, že užitek peněz není lineární.
- Nechť mám v tuto chvíli k peněz. Užitek mít o n víc je zhruba (v dolarech):

$$U(S_{k+n}) = -263.31 + 22.09 log(n + 150000)$$

v rozmezí –\$150000 a \$800000. (Mr. Beard)

Užitek peněz

Loterie

Opět výběr ze dvou loterií:

- Buď dostat \$1000000
- nebo hodit korunou a na 50% dostat \$3000000

Užitek peněz

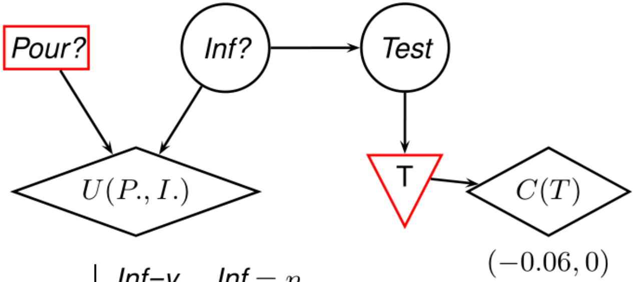
- Ekonomové říkají, že užitek peněz není lineární.
- Nechť mám v tuto chvíli k peněz. Užitek mít o n víc je zhruba (v dolarech):

$$U(S_{k+n}) = -263.31 + 22.09 log(n + 150000)$$

v rozmezí –\$150000 a \$800000. (Mr. Beard)

Příklad - Mléko

Nadojené mléko: Při uvedených cenách: použít, vyhodit, testovat?



	Inf=*y*	*Inf* $= n$
Pour?=*y*	0	100
Pour?=*n*	98	98

Očekávaný zisk (expected utility, value)

I když známe zisk $V(d, x, e)$ v každé možné situaci d, x, e, nemáme jistotu, která situace nastane. Proto nemůžeme maximalizovat $V(d, x, e)$, ale maximalizujeme **očekávaný zisk**:

$$EU(d|e) = \sum_x V(d, x, e) \cdot P(X|d, e)$$

Pokud máme více funkcí užitku $V_1, \ldots, V_n$, pak (téměř vždy) uvažujeme aditivní skládání, tj. $V(U) = V_1(U) + \ldots + V_n(U)$.
Jednotlivé funkce $V_i(U)$ mohou záviset na různých podmnožinách prostoru náhodných veličin U.

(Pravděpodobnostní) rozhodovací stromy

- Rozhodovací strom obsahuje tři typy uzlů:
  - rozhodnutí D_i
  - náhodné uzly X_i
  - listy; listům je přiřazen zisk v příslušné situaci definované cestou od kořene.
- pro každou možnou hodnotu rozhodnutí D a náhodné veličiny X vychází z uzlu D resp. X jedna hrana označená tímto výsledkem d_j, x_j
- větve z náhodné veličiny X navíc označíme pravděpodobností, s jakou větev zvolíme, za podmínky, že se do uzlu X dostaneme, tj. $P(X = x_j | \text{cesta z kořene do X})$.
- na cestě od kořene do listu se každé rozhodnutí a náhodná veličina vyskytuje nejvýše jednou (nemusí se vyskytovat, pokud neovlivňuje užitek ani pravděpodobnost dosažení listu),
- **cesta odpovídá časovému uspořádání**: náhodná veličina je před rozhodnutím právě když je hodnota této veličiny známa před rozhodnutím.

(Pravděpodobnostní) rozhodovací stromy

- Rozhodovací strom obsahuje tři typy uzlů:
  - rozhodnutí D_i
  - náhodné uzly X_j
  - listy; listům je přiřazen zisk v příslušné situaci definované cestou od kořene.
- pro každou možnou hodnotu rozhodnutí D a náhodné veličiny X vychází z uzlu D resp. X jedna hrana označená tímto výsledkem d_j, x_j
- větve z náhodné veličiny X navíc označíme pravděpodobností, s jakou větev zvolíme, za podmínky, že se do uzlu X dostaneme, tj. $P(X = x_j | \text{cesta z kořene do X})$.
- na cestě od kořene do listu se každé rozhodnutí a náhodná veličina vyskytuje nejvýše jednou (nemusí se vyskytovat, pokud neovlivňuje užitek ani pravděpodobnost dosažení listu),
- **cesta odpovídá časovému uspořádání**: náhodná veličina je před rozhodnutím právě když je hodnota této veličiny známa před rozhodnutím.

(Pravděpodobnostní) rozhodovací stromy

- Rozhodovací strom obsahuje tři typy uzlů:
  - rozhodnutí D_i
  - náhodné uzly X_j
  - listy; listům je přiřazen zisk v příslušné situaci definované cestou od kořene.
- pro každou možnou hodnotu rozhodnutí D a náhodné veličiny X vychází z uzlu D resp. X jedna hrana označená tímto výsledkem d_j, x_j
- větve z náhodné veličiny X navíc označíme pravděpodobností, s jakou větev zvolíme, za podmínky, že se do uzlu X dostaneme, tj. $P(X = x_j | \text{cesta z kořene do X})$.
- na cestě od kořene do listu se každé rozhodnutí a náhodná veličina vyskytuje nejvýše jednou (nemusí se vyskytovat, pokud neovlivňuje užitek ani pravděpodobnost dosažení listu),
- **cesta odpovídá časovému uspořádání**: náhodná veličina je před rozhodnutím právě když je hodnota této veličiny známa před rozhodnutím.

(Pravděpodobnostní) rozhodovací stromy

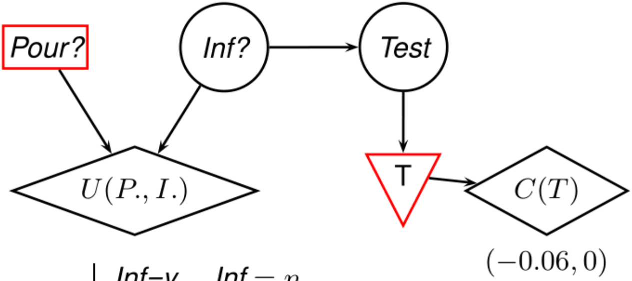
- Rozhodovací strom obsahuje tři typy uzlů:
  - rozhodnutí D_i
  - náhodné uzly X_j
  - listy; listům je přiřazen zisk v příslušné situaci definované cestou od kořene.
- pro každou možnou hodnotu rozhodnutí D a náhodné veličiny X vychází z uzlu D resp. X jedna hrana označená tímto výsledkem d_j, x_j
- větve z náhodné veličiny X navíc označíme pravděpodobností, s jakou větev zvolíme, za podmínky, že se do uzlu X dostaneme, tj. $P(X = x_j | \text{cesta z kořene do X})$.
- na cestě od kořene do listu se každé rozhodnutí a náhodná veličina vyskytuje nejvýše jednou (nemusí se vyskytovat, pokud neovlivňuje užitek ani pravděpodobnost dosažení listu),
- **cesta odpovídá časovému uspořádání**: náhodná veličina je před rozhodnutím právě když je hodnota této veličiny známa před rozhodnutím.

(Pravděpodobnostní) rozhodovací stromy

- Rozhodovací strom obsahuje tři typy uzlů:
  - rozhodnutí D_i
  - náhodné uzly X_j
  - listy; listům je přiřazen zisk v příslušné situaci definované cestou od kořene.
- pro každou možnou hodnotu rozhodnutí D a náhodné veličiny X vychází z uzlu D resp. X jedna hrana označená tímto výsledkem d_j, x_j
- větve z náhodné veličiny X navíc označíme pravděpodobností, s jakou větev zvolíme, za podmínky, že se do uzlu X dostaneme, tj. $P(X = x_j | \text{cesta z kořene do X})$.
- na cestě od kořene do listu se každé rozhodnutí a náhodná veličina vyskytuje nejvýše jednou (nemusí se vyskytovat, pokud neovlivňuje užitek ani pravděpodobnost dosažení listu),
- **cesta odpovídá časovému uspořádání**: náhodná veličina je před rozhodnutím právě když je hodnota této veličiny známa před rozhodnutím.

Příklad - Mléko

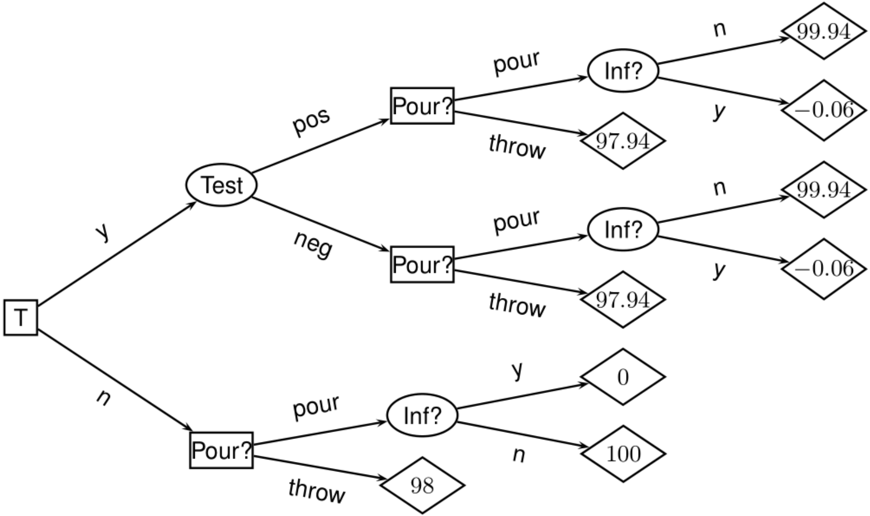
Nadojené mléko: Při uvedených cenách: použít, vyhodit, testovat?



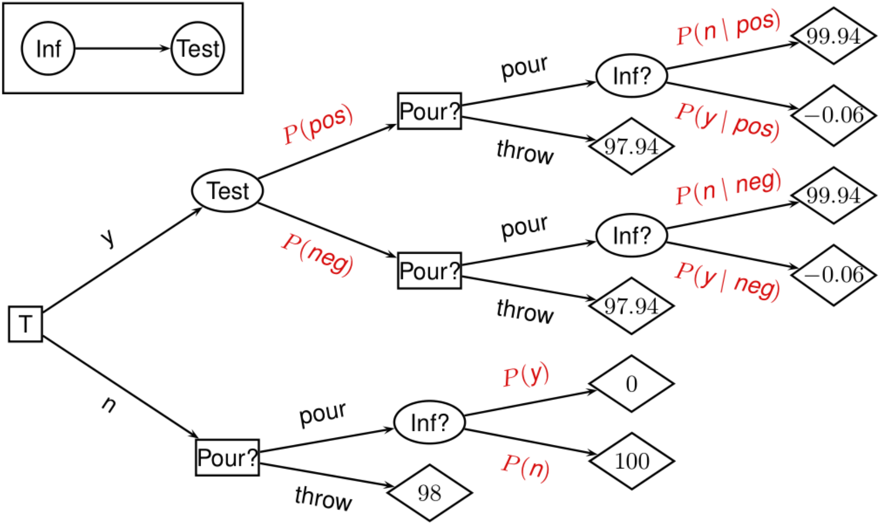
	Inf=*y*	*Inf* $= n$
Pour?=*y*	0	100
Pour?=*n*	98	98

Příklad - Mléko

Nadojené mléko: použít, vyhodit, testovat?



Pravděpodobnostní model + ceny (užitek).

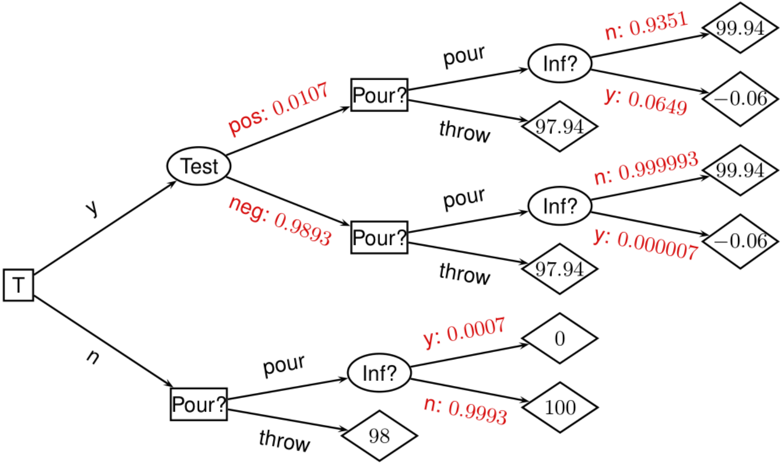


Příklad - Mléko

Dopočteme pravděpodobnosti.

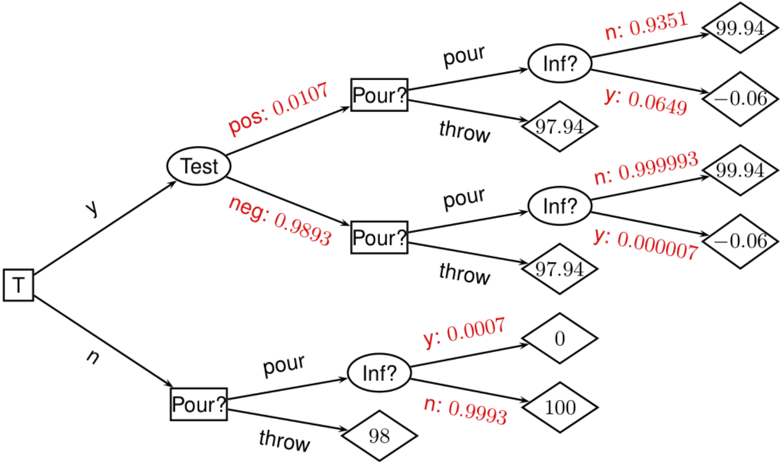
inf <- cptable(~inf, values=c(0.0007,0.9993),levels=c('yes','no'))

test <- cptable(~test+inf, values=c(1,99,99,1),levels=c('pos','neg'))



Příklad - Mléko

Vyhodnotíme. Vážený součet v náhodných uzlech, maximum v rozhodnutích.
$EU(dole) = 0 \cdot P(y) + 100 * P(n) = 99.93$,
$EU(nahore) = 97.94 \cdot P(pos) + 99.94 \cdot P(n, neg) - 0.06 \cdot P(y, neg) = 99.92$.



Vyhodnocení rozhodovacího stromu

- Vyhodnocení začínáme od listů, kde je znám zisk $EU(list) = V(list)$.
- Jsou–li všechny děti uzlu vyhodnoceny, vyhodnotíme uzel:
  - v **rozhodovacím uzlu** vybereme dítě s nejvyšším ziskem a opíšeme jeho hodnotu; větev označíme.
  - v **náhodném uzlu** spočteme $EU(X|cesta) = \sum_{deti} EU(dite|cesta) \cdot P(dite|cesta)$.
  - v kořeni nám vyjde očekávaný zisk **optimální strategie**.

Definition

Strategie řešení rozhodování za nejistoty je předpis (např. rozhodovací strom) jaké rozhodnutí zvolit za každé možné situace předchozích rozhodnutí a pozorovaných náhodných veličin.
Optimální strategie je strategie s maximálním očekávaným ziskem ze všech strategií pro daný problém.

Influenční diagram (Influence diagram, Decision graph)

Influenční diagram je DAG (orientovaný graf bez cyklů) se třemi typy uzlů a dvěma typy tabulek:

- Rozhodovací a náhodné uzly mají konečné množiny vzájemně se vylučujících stavů, uzly užitku nemají žádné stavy.
- Obdélníkové **rozhodovací uzly** nemají přiřazenou tabulku, resp. tabulku podkladů pro rozhodnutí vypočteme
- Elipsové **náhodné uzly** mají přiřazeny podmíněnou pravděpodobnost stavu pro možné konfigurace rodičů, stejně jak bylo v bayes. síti
- Kosočtvercové **uzly užitku**, (**utility**) mají přiřazenu výplatu (reálné číslo) pro každou konfiguraci rodičů.
- Hrany do náhodných veličin mají závislostní význam, jako v BN.
- **Hrany do rozhodovacích uzlů** z náhodných veličin mají význam informační:
- Předpokládáme **nezapomínání** – pokud předchozí rozhodnutí znalo veličinu A, pak ji zná i následné rozhodnutí.
- **Influenční diagram vyžaduje orientovanou cestu přes všechna rozhodnutí**, tj. časové uspořádání na rozhodnutích. (Nevadí, když jsou na cestě i náhodné uzly).

Influenční diagram (Influence diagram, Decision graph)

Influenční diagram je DAG (orientovaný graf bez cyklů) se třemi typy uzlů a dvěma typy tabulek:

- Rozhodovací a náhodné uzly mají konečné množiny vzájemně se vylučujících stavů, uzly užitku nemají žádné stavy.
- Obdélníkové **rozhodovací uzly** nemají přiřazenou tabulku, resp. tabulku podkladů pro rozhodnutí vypočteme
- Elipsové **náhodné uzly** mají přiřazeny podmíněnou pravděpodobnost stavu pro možné konfigurace rodičů, stejně jak bylo v bayes. síti
- Kosočtvercové **uzly užitku**, (**utility**) mají přiřazenu výplatu (reálné číslo) pro každou konfiguraci rodičů.
- Hrany do náhodných veličin mají závislostní význam, jako v BN.
- **Hrany do rozhodovacích uzlů** z náhodných veličin mají význam informační:
- Předpokládáme **nezapomínání** – pokud předchozí rozhodnutí znalo veličinu A, pak ji zná i následné rozhodnutí.
- **Influenční diagram vyžaduje orientovanou cestu přes všechna rozhodnutí**, tj. časové uspořádání na rozhodnutích. (Nevadí, když jsou na cestě i náhodné uzly).

Influenční diagram (Influence diagram, Decision graph)

Influenční diagram je DAG (orientovaný graf bez cyklů) se třemi typy uzlů a dvěma typy tabulek:

- Rozhodovací a náhodné uzly mají konečné množiny vzájemně se vylučujících stavů, uzly užitku nemají žádné stavy.
- Obdélníkové **rozhodovací uzly** nemají přiřazenou tabulku, resp. tabulku podkladů pro rozhodnutí vypočteme
- Elipsové **náhodné uzly** mají přiřazeny podmíněnou pravděpodobnost stavu pro možné konfigurace rodičů, stejně jak bylo v bayes. síti
- Kosočtvercové **uzly užitku**, (**utility**) mají přiřazenu výplatu (reálné číslo) pro každou konfiguraci rodičů.
- Hrany do náhodných veličin mají závislostní význam, jako v BN.
- **Hrany do rozhodovacích uzlů** z náhodných veličin mají význam informační.
- Předpokládáme **nezapomínání** – pokud předchozí rozhodnutí znalo veličinu A, pak ji zná i následné rozhodnutí.
- **Influenční diagram vyžaduje orientovanou cestu přes všechna rozhodnutí**, tj. časové uspořádání na rozhodnutích. (Nevadí, když jsou na cestě i náhodné uzly).

Influenční diagram (Influence diagram, Decision graph)

Influenční diagram je DAG (orientovaný graf bez cyklů) se třemi typy uzlů a dvěma typy tabulek:

- Rozhodovací a náhodné uzly mají konečné množiny vzájemně se vylučujících stavů, uzly užitku nemají žádné stavy.
- Obdélníkové **rozhodovací uzly** nemají přiřazenou tabulku, resp. tabulku podkladů pro rozhodnutí vypočteme
- Elipsové **náhodné uzly** mají přiřazeny podmíněnou pravděpodobnost stavu pro možné konfigurace rodičů, stejně jak bylo v bayes. síti
- Kosočtvercové **uzly užitku**, (**utility**) mají přiřazenu výplatu (reálné číslo) pro každou konfiguraci rodičů.
- Hrany do náhodných veličin mají závislostní význam, jako v BN.
- **Hrany do rozhodovacích uzlů** z náhodných veličin mají význam informační:
- Předpokládáme **nezapomínání** – pokud předchozí rozhodnutí znalo veličinu A, pak ji zná i následné rozhodnutí.
- **Influenční diagram vyžaduje orientovanou cestu přes všechna rozhodnutí**, tj. časové uspořádání na rozhodnutích. (Nevadí, když jsou na cestě i náhodné uzly).

Influenční diagram (Influence diagram, Decision graph)

Influenční diagram je DAG (orientovaný graf bez cyklů) se třemi typy uzlů a dvěma typy tabulek:

- Rozhodovací a náhodné uzly mají konečné množiny vzájemně se vylučujících stavů, uzly užitku nemají žádné stavy.
- Obdélníkové **rozhodovací uzly** nemají přiřazenou tabulku, resp. tabulku podkladů pro rozhodnutí vypočteme
- Elipsové **náhodné uzly** mají přiřazeny podmíněnou pravděpodobnost stavu pro možné konfigurace rodičů, stejně jak bylo v bayes. síti
- Kosočtvercové **uzly užitku**, (**utility**) mají přiřazenu výplatu (reálné číslo) pro každou konfiguraci rodičů.
- Hrany do náhodných veličin mají závislostní význam, jako v BN.
- **Hrany do rozhodovacích uzlů** z náhodných veličin mají význam informační:
- Předpokládáme **nezapomínání** – pokud předchozí rozhodnutí znalo veličinu A, pak ji zná i následné rozhodnutí.
- **Influenční diagram vyžaduje orientovanou cestu přes všechna rozhodnutí**, tj. časové uspořádání na rozhodnutích. (Nevadí, když jsou na cestě i náhodné uzly).

Influenční diagram (Influence diagram, Decision graph)

Influenční diagram je DAG (orientovaný graf bez cyklů) se třemi typy uzlů a dvěma typy tabulek:

- Rozhodovací a náhodné uzly mají konečné množiny vzájemně se vylučujících stavů, uzly užitku nemají žádné stavy.
- Obdélníkové **rozhodovací uzly** nemají přiřazenou tabulku, resp. tabulku podkladů pro rozhodnutí vypočteme
- Elipsové **náhodné uzly** mají přiřazeny podmíněnou pravděpodobnost stavu pro možné konfigurace rodičů, stejně jak bylo v bayes. síti
- Kosočtvercové **uzly užitku**, (**utility**) mají přiřazenu výplatu (reálné číslo) pro každou konfiguraci rodičů.
- Hrany do náhodných veličin mají závislostní význam, jako v BN.
- **Hrany do rozhodovacích uzlů** z náhodných veličin mají význam informační: před rozhodnutím známe stav příslušné náhodné veličiny;
  z rozhodovacích uzlů mají význam časového uspořádání – rozhodnutí, ze kterého jde hrana, musí být uskutečněno dřív, než rozhodnutí – dítě.
- Předpokládáme **nezapomínání** – pokud předchozí rozhodnutí znalo veličinu A, pak ji zná i následné rozhodnutí.
- **Influenční diagram vyžaduje orientovanou cestu přes všechna rozhodnutí**, tj. časové uspořádání na rozhodnutích. (Nevadí, když jsou na

Influenční diagram (Influence diagram, Decision graph)

Influenční diagram je DAG (orientovaný graf bez cyklů) se třemi typy uzlů a dvěma typy tabulek:

- Rozhodovací a náhodné uzly mají konečné množiny vzájemně se vylučujících stavů, uzly užitku nemají žádné stavy.
- Obdélníkové **rozhodovací uzly** nemají přiřazenou tabulku, resp. tabulku podkladů pro rozhodnutí vypočteme
- Elipsové **náhodné uzly** mají přiřazeny podmíněnou pravděpodobnost stavu pro možné konfigurace rodičů, stejně jak bylo v bayes. síti
- Kosočtvercové **uzly užitku**, (**utility**) mají přiřazenu výplatu (reálné číslo) pro každou konfiguraci rodičů.
- Hrany do náhodných veličin mají závislostní význam, jako v BN.
- **Hrany do rozhodovacích uzlů** z náhodných veličin mají význam informační:
- Předpokládáme **nezapomínání** – pokud předchozí rozhodnutí znalo veličinu A, pak ji zná i následné rozhodnutí.
- **Influenční diagram vyžaduje orientovanou cestu přes všechna rozhodnutí**, tj. časové uspořádání na rozhodnutích. (Nevadí, když jsou na cestě i náhodné uzly).

Influenční diagram (Influence diagram, Decision graph)

Influenční diagram je DAG (orientovaný graf bez cyklů) se třemi typy uzlů a dvěma typy tabulek:

- Rozhodovací a náhodné uzly mají konečné množiny vzájemně se vylučujících stavů, uzly užitku nemají žádné stavy.
- Obdélníkové **rozhodovací uzly** nemají přiřazenou tabulku, resp. tabulku podkladů pro rozhodnutí vypočteme
- Elipsové **náhodné uzly** mají přiřazeny podmíněnou pravděpodobnost stavu pro možné konfigurace rodičů, stejně jak bylo v bayes. síti
- Kosočtvercové **uzly užitku**, (**utility**) mají přiřazenu výplatu (reálné číslo) pro každou konfiguraci rodičů.
- Hrany do náhodných veličin mají závislostní význam, jako v BN.
- **Hrany do rozhodovacích uzlů** z náhodných veličin mají význam informační:
- Předpokládáme **nezapomínání** – pokud předchozí rozhodnutí znalo veličinu A, pak ji zná i následné rozhodnutí.
- **Influenční diagram vyžaduje orientovanou cestu přes všechna rozhodnutí**, tj. časové uspořádání na rozhodnutích. (Nevadí, když jsou na cestě i náhodné uzly).

Příklad - Poker (T.D.Nielsen, kniha)

- Každý dostane 5 karet
- poprvé: nejvýše 3 může vyměnit
- podruhé: nejvýše 2 může vyměnit
- můžu 'vsadit' nebo vzdát
- vyšší karty berou talon.

Úkol

Navrhněte model - rozhodovací graf.

Vyhodnocení influenčního diagramu

Jedna cesta je vytvořit rozhodovací strom a ten vyhodnotit.
Influenční diagram vyžaduje **časové uspořádání rozhodnutí**, nechť máme provádět rozhodnutí v pořadí $D_1, \ldots, D_n$. Označíme I_0 množinu náhodných veličin, které může D_1 pozorovat; obecně I_i bude množina pozorování, které může pozorovat D_{i+1}, ale ne D_i. I_n jsou pozorování, které nezná ani poslední rozhodnutí. Tím dostáváme **částečné časové uspořádání** rozhodovacích a náhodných uzlů $I_0 < D_1 < I_1 < \ldots < D_n < I_n$. Při konstrukci stromu musí být vždy "menší"uzly blíže kořeni.

Řetězové pravidlo pro influenční diagramy (ID)

Nechť jsou v ID náhodné veličiny $\mathcal{O}$ a rozhodnutí $D_1, \ldots, D_n$. Pak:

$$P(\mathcal{O}|D_1, \ldots, D_n) = \Pi_{X \in \mathcal{O}} P(X|pa(X)).$$

Na základě tohoto pravidla můžeme vypočítat všechny pravděpodobnosti potřebné v rozhodovacím stromě.

Optimální stragetie

Při daném časovém uspořádání $I_0 < D_1 < I_1 < \ldots < D_n < I_n$ je optimální strategie pro D_i:

$$\sigma_i(I_0, D_1, I_1, \ldots, D_{i-1}, I_{i-1}) =$$

$$argmax_{D_i} \sum_{I_i} max_{D_{i+1}} \ldots max_{D_n} \sum_{I_n} P(U|D_1, \ldots, D_n) V(U, D_1, \ldots, D_n)$$

Očekávaná hodnota stragetie počínající v D_i je:

$$\rho_i(I_0, D_1, I_1, \ldots, D_{i-1}, I_{i-1}) = \frac{1}{P(I_0, \ldots, I_{i-1}|D_1, \ldots, D_{i-1})} \cdot$$

$$max_{D_i} \sum_{I_i} max_{D_{i+1}} \ldots max_{D_n} \sum_{I_n} P(U|D_1, \ldots, D_n) V(U, D_1, \ldots, D_n)$$

Algoritmus eliminace proměnných

- Do seznamu Φ_0 dáme všechny pravděpodobnostní tabulky $P(A_i|pa(A_i))$.
- Do seznamu Ψ_0 dáme všechny tabulky užitku $V_j(pa(V_j))$.
- Postupně budeme eliminovat **proti směru časového uspořádání** všechny proměnné. Při eliminaci rozhodnutí si zapamatujeme optimální strategii tohoto rozhodnutí.
- Při použití vyhodnoceného ID nejdříve zadáme hodnoty pozorované před D_1, pak zvolíme d_1 s nejvetším očekávaným ziskem. Pokračujeme dále zadávání pozorovaných hodnot a volbou rozhodnutí, dokud nedojdeme k poslednímu rozhodnutí.

Algoritmus eliminace proměnných

- Do seznamu Φ_0 dáme všechny pravděpodobnostní tabulky $P(A_i|pa(A_i))$.
- Do seznamu Ψ_0 dáme všechny tabulky užitku $V_j(pa(V_j))$.
- Postupně budeme eliminovat **proti směru časového uspořádání** všechny proměnné. Při eliminaci rozhodnutí si zapamatujeme optimální strategii tohoto rozhodnutí.
- Při použití vyhodnoceného ID nejdříve zadáme hodnoty pozorované před D_1, pak zvolíme d_1 s nejvetším očekávaným ziskem. Pokračujeme dále zadávání pozorovaných hodnot a volbou rozhodnutí, dokud nedojdeme k poslednímu rozhodnutí.

Algoritmus eliminace proměnných

- Do seznamu Φ_0 dáme všechny pravděpodobnostní tabulky $P(A_i|pa(A_i))$.
- Do seznamu Ψ_0 dáme všechny tabulky užitku $V_j(pa(V_j))$.
- Postupně budeme eliminovat **proti směru časového uspořádání** všechny proměnné. Při eliminaci rozhodnutí si zapamatujeme optimální strategii tohoto rozhodnutí.
- Při použití vyhodnoceného ID nejdříve zadáme hodnoty pozorované před D_1, pak zvolíme d_1 s nejvetším očekávaným ziskem. Pokračujeme dále zadávání pozorovaných hodnot a volbou rozhodnutí, dokud nedojdeme k poslednímu rozhodnutí.

Algoritmus eliminace proměnných

- Do seznamu Φ_0 dáme všechny pravděpodobnostní tabulky $P(A_i|pa(A_i))$.
- Do seznamu Ψ_0 dáme všechny tabulky užitku $V_j(pa(V_j))$.
- Postupně budeme eliminovat **proti směru časového uspořádání** všechny proměnné. Při eliminaci rozhodnutí si zapamatujeme optimální strategii tohoto rozhodnutí.
- Při použití vyhodnoceného ID nejdříve zadáme hodnoty pozorované před D_1, pak zvolíme d_1 s nejvetším očekávaným ziskem. Pokračujeme dále zadávání pozorovaných hodnot a volbou rozhodnutí, dokud nedojdeme k poslednímu rozhodnutí.

Eliminace proměnné X v kroku i znamená:

1. $\Phi_X = \{\phi \in \Phi_{i-1} | X \in dom(\phi)\}$
   $\Psi_X = \{\psi \in \Psi_{i-1} | X \in dom(\psi)\}$
2. Je–li X **náhodná veličina**
   $\phi_X = \sum_X \prod \Phi_X$
   $\psi_X = \frac{1}{\phi_X} \sum_X \prod \Phi_X \left(\sum \Psi_X\right)$
3. jinak X **rozhodnutí**
   $\phi_X = max_X \prod \Phi_X$
   $\psi_X = max_X \left(\sum \Psi_X\right)$
4. každopádně:
   $\Phi_i = \Phi_{i-1} \setminus \Phi_X \cup \{\phi_X\}$
   $\Psi_i = \Psi_{i-1} \setminus \Psi_X \cup \{\psi_X\}$

Varianty ID

- LIMIDs - Limited Memory IDs
- UIDs - Unconstrained IDs
- a mnoho jazyků na asymetrické problémy (např. Valuation networks,AIDs)
- CEG - Chain Event Graphs - blíže k rozhodovacím stromům, ale snaží se je maximálně spojovat 'coalescencí'.

Opakování v čase

Příklad: Rybolov

Variaty ID

- LIMIDs - Limited Memory IDs
- UIDs - Unconstrained IDs
- a mnoho jazyků na asymetrické problémy (např. Valuation networks,AIDs)
- CEG - Chain Event Graphs - blíže k rozhodovacím stromům, ale snaží se je maximálně spojovat 'coalescencí'.

Opakování v čase

Příklad: Rybolov

Varianty ID

- LIMIDs - Limited Memory IDs
- UIDs - Unconstrained IDs
- a mnoho jazyků na asymetrické problémy (např. Valuation networks,AIDs)
- CEG - Chain Event Graphs - blíže k rozhodovacím stromům, ale snaží se je maximálně spojovat 'coalescencí'.

Opakování v čase

Příklad: Rybolov

Varianty ID

- LIMIDs - Limited Memory IDs
- UIDs - Unconstrained IDs
- a mnoho jazyků na asymetrické problémy (např. Valuation networks,AIDs)
- CEG - Chain Event Graphs - blíže k rozhodovacím stromům, ale snaží se je maximálně spojovat 'coalescencí'.

Opakování v čase

Příklad: Rybolov

Variianty ID

- LIMIDs - Limited Memory IDs
- UIDs - Unconstrained IDs
- a mnoho jazyků na asymetrické problémy (např. Valuation networks,AIDs)
- CEG - Chain Event Graphs - blíže k rozhodovacím stromům, ale snaží se je maximálně spojovat 'coalescencí'.

Opakování v čase

Příklad: Rybolov

Markovské rozhodovací procesy

- Předpokládáme, že se množina možných stavů S nemění v průběhu času
- **Markovská vlastnost** stav v čase $t+1$ je nezávislý na stavu v čase $t-i$, $i > 0$ při znalosti stavu v čase t, tj.

$$S_{t+1} \perp\!\!\!\perp S_{t-i} | S_t$$

- Toto jsou plně pozorované markovské procesy prvního řádu; markovské procesy vyššího řádu dovolují závislost na více předchozích stavech.

Markovské rozhodovací procesy

- Předpokládáme, že se množina možných stavů S nemění v průběhu času
- **Markovská vlastnost** stav v čase $t+1$ je nezávislý na stavu v čase $t-i$, $i > 0$ při znalosti stavu v čase t, tj.

$$S_{t+1} \perp\!\!\!\perp S_{t-i} | S_t$$

- Toto jsou plně pozorované markovské procesy prvního řádu; markovské procesy vyššího řádu dovolují závislost na více předchozích stavech.

Markovské rozhodovací procesy

- Předpokládáme, že se množina možných stavů S nemění v průběhu času
- **Markovská vlastnost** stav v čase $t+1$ je nezávislý na stavu v čase $t-i$, $i > 0$ při znalosti stavu v čase t, tj.

$$S_{t+1} \perp\!\!\!\perp S_{t-i} | S_t$$

- Toto jsou plně pozorované markovské procesy prvního řádu; markovské procesy vyššího řádu dovolují závislost na více předchozích stavech.

Markovský rozhodovací proces MDP

- Množina stavů S v každém časovém bodě stejná
- Počáteční stav s_0
- Množina možných akcí A
- Matice přechodu $T(s, a, s^{|})$
- Výplata v každém stavu $R(s)$

Kumulovaná výplata

Protože jde o proces v čase, výplaty musíme sčítat.
Jsou dvě možnosti:

- MDP s konečným horizontem – předem stanoví počet kroků
- nekonečný proces a součet, ale hodnoty v budoucnosti budeme počítat s nižší hodnotou (kdo ví, co bude pak).
  Zvolíme tzv. **diskontní faktor** γ, $0 < \gamma < 1$, a maximalizujeme

$$U(s_0, \ldots, s_t, \ldots) = \sum_{t=0}^{\infty} \gamma^t R(s_t)$$

- γ odpovídá úrokové míře $\frac{1}{\gamma} - 1$, kterou musíme platit.
- Vzhledem ke konečnosti S je $U(s_0, \ldots, s_t, \ldots) \leq \frac{R_{max}}{(1-\gamma)}$

Kumulovaná výplata

Protože jde o proces v čase, výplaty musíme sčítat.
Jsou dvě možnosti:

- MDP s konečným horizontem – předem stanoví počet kroků
- nekonečný proces a součet, ale hodnoty v budoucnosti budeme počítat s nižší hodnotou (kdo ví, co bude pak).
  Zvolíme tzv. **diskontní faktor** γ, $0 < \gamma < 1$, a maximalizujeme

$$U(s_0, \ldots, s_t, \ldots) = \sum_{t=0}^{\infty} \gamma^t R(s_t)$$

- γ odpovídá úrokové míře $\frac{1}{\gamma} - 1$, kterou musíme platit.
- Vzhledem ke konečnosti S je $U(s_0, \ldots, s_t, \ldots) \leq \frac{R_{max}}{(1-\gamma)}$

Kumulovaná výplata

Protože jde o proces v čase, výplaty musíme sčítat.
Jsou dvě možnosti:

- MDP s konečným horizontem – předem stanoví počet kroků
- nekonečný proces a součet, ale hodnoty v budoucnosti budeme počítat s nižší hodnotou (kdo ví, co bude pak).
  Zvolíme tzv. **diskontní faktor** γ, $0 < \gamma < 1$, a maximalizujeme

$$U(s_0, \ldots, s_t, \ldots) = \sum_{t=0}^{\infty} \gamma^t R(s_t)$$

- γ odpovídá úrokové míře $\frac{1}{\gamma} - 1$, kterou musíme platit.
- Vzhledem ke konečnosti S je $U(s_0, \ldots, s_t, \ldots) \leq \frac{R_{max}}{(1-\gamma)}$

Kumulovaná výplata

Protože jde o proces v čase, výplaty musíme sčítat.
Jsou dvě možnosti:

- MDP s konečným horizontem – předem stanoví počet kroků
- nekonečný proces a součet, ale hodnoty v budoucnosti budeme počítat s nižší hodnotou (kdo ví, co bude pak).
  Zvolíme tzv. **diskontní faktor** γ, $0 < \gamma < 1$, a maximalizujeme

$$U(s_0, \ldots, s_t, \ldots) = \sum_{t=0}^{\infty} \gamma^t R(s_t)$$

- γ odpovídá úrokové míře $\frac{1}{\gamma} - 1$, kterou musíme platit.
- Vzhledem ke konečnosti S je $U(s_0, \ldots, s_t, \ldots) \leq \frac{R_{max}}{(1-\gamma)}$

Strategie (policy)

Cílem je najít takovou **strategii** π^*, která maximalizuje očekávanou výplatu, tj.

$$\pi^* = argmax_\pi E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t)|\pi\right]$$

- Konečný horizont vede k **nestacionární** strategii, tj. různé strategii dle toho, kolik času zbývá do konce.
- Nekonečný horizont vede ke **stacionární** strategii, tj. doporučený tah v políčku (stavu) je stejný nezávisle na počtu již provedených tahů.
- Reprezentovat stacionární strategii je snažší.
- Máme–li jistotu, že agent musí skončit v cílovém stavu, můžeme počítat s $\gamma = 1$, jinak bychom mohli neustále zvyšovat užitek a neměli bychom zajištěno nalezení optimální strategie. (Šlo by hledat maximální zisk za k kroků.)

Strategie (policy)

Cílem je najít takovou **strategii** π^*, která maximalizuje očekávanou výplatu, tj.

$$\pi^* = argmax_\pi E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t)|\pi\right]$$

- Konečný horizont vede k **nestacionární** strategii, tj. různé strategii dle toho, kolik času zbývá do konce.
- Nekonečný horizont vede ke **stacionární** strategii, tj. doporučený tah v políčku (stavu) je stejný nezávisle na počtu již provedených tahů.
- Reprezentovat stacionární strategii je snažší.
- Máme–li jistotu, že agent musí skončit v cílovém stavu, můžeme počítat s $\gamma = 1$, jinak bychom mohli neustále zvyšovat užitek a neměli bychom zajištěno nalezení optimální strategie. (Šlo by hledat maximální zisk za k kroků.)

Strategie (policy)

Cílem je najít takovou **strategii** π^*, která maximalizuje očekávanou výplatu, tj.

$$\pi^* = argmax_\pi E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t)|\pi\right]$$

- Konečný horizont vede k **nestacionární** strategii, tj. různé strategii dle toho, kolik času zbývá do konce.
- Nekonečný horizont vede ke **stacionární** strategii, tj. doporučený tah v políčku (stavu) je stejný nezávisle na počtu již provedených tahů.
- Reprezentovat stacionární strategii je snažší.
- Máme–li jistotu, že agent musí skončit v cílovém stavu, můžeme počítat s $\gamma = 1$, jinak bychom mohli neustále zvyšovat užitek a neměli bychom zajištěno nalezení optimální strategie. (Šlo by hledat maximální zisk za k kroků.)

Strategie (policy)

Cílem je najít takovou **strategii** π^*, která maximalizuje očekávanou výplatu, tj.

$$\pi^* = argmax_\pi E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t)|\pi\right]$$

- Konečný horizont vede k **nestacionární** strategii, tj. různé strategii dle toho, kolik času zbývá do konce.
- Nekonečný horizont vede ke **stacionární** strategii, tj. doporučený tah v políčku (stavu) je stejný nezávisle na počtu již provedených tahů.
- Reprezentovat stacionární strategii je snažší.
- Máme–li jistotu, že agent musí skončit v cílovém stavu, můžeme počítat s $\gamma = 1$, jinak bychom mohli neustále zvyšovat užitek a neměli bychom zajištěno nalezení optimální strategie. (Šlo by hledat maximální zisk za k kroků.)

Algoritmus řešení MDP Iterací hodnot

input: MDP, stavy S, přechody T, výplata R, diskontní f. γ,ϵ
prom.: $U, U^{|}$, vektory užitku stavů v S, na poč. 0
δ maximální změna užitku v jedné iteraci

repeat
$U \leftarrow U^{|}$; $\delta \leftarrow 0$
for each state s in S **do**
$U^{|}[s] \leftarrow R[s] + \gamma max_a \sum_{s^{|}} T(s, a, s^{|}) U(s^{|})$
if $|U^{|}[s] - U[s]| > \delta$ **then** $\delta \leftarrow |U^{|}[s] - U[s]|$
until $\delta < \epsilon(1-\gamma)/\gamma$
return U

Iterace strategie (policy iteration)

input: MDP, stavy S, přechody T, výplata R, diskontní f. γ,ϵ
prom.: U, vektory užitku stavů v S, na poč. 0
π strategie, na začátku náhodná

repeat
$U \leftarrow HODNOTA\_STRATEGIE(\pi, U, MPD)$
$unchanged? \leftarrow true$
for each state s in S **do**
if $max_a \sum_{s^|} T(s, a, s^|)U[s^|] > \sum_{s^|} T(s, \pi[s], s^|)U[s^|]$
then
$\pi[s] \leftarrow argmax_a \sum_{s^|} T(s, a, s^|)U[s^|]$
$unchanged? \leftarrow false$
until $unchanged?$
return π

Výpočet $HODNOTA\_STRATEGIE(\pi, U, MPD)$ vyžaduje vyřešení soustavy $|S|$ lineárních rovnic pro $U[s]$ v jednotlivých stavech.

Částečně pozorovatelné markovské procesy (POMDP)

- Nejsme schopni pozorovat stav, pouze skrze nepřesné senzory.
- Bude příště.
- Pointy:
  - Proces je opět makrovský, pokud bereme BELIEF (PRAVDĚPODOBNOSTNÍ ROZLOŽENÍ) na stavech. Těch je ale nekonečně, tedy máme problém.
  - Tzv. Witness algoritmus – hledáme svědka, že naše strategie není optimální.