Decision tree analysis with SilverDecisions.pl

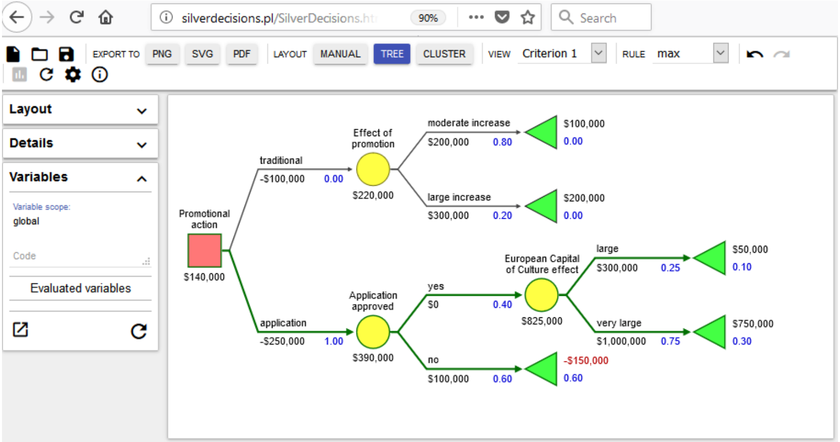
Bogumił Kamiński and Michał Jakubczyk
and Przemysław Szufel

SGH - Warsaw School of Economics

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645860

May 16, 2018

SilverDecisions.pl user interface



Decision tree

A decision tree DT is a tuple (G, y, p)

- Directed graph $G = (V, E)$, $E \subset V^2$, with set of nodes V split into three disjoint sets $V = \mathcal{D} \cup \mathcal{C} \cup \mathcal{T}$
  - $\mathcal{D}$ decision node
  - $\mathcal{C}$ chance node
  - $\mathcal{T}$ terminal node
  - for each edge $e \in E$
    - $e_1 \in V$ denote its first element (parent node)
    - $e_2 \in V$ denote its second element (child node)
- payoff function: $y \colon E \to \mathbb{R}$
- probabilities function: $p \colon \{e \in E : e_1 \in \mathcal{C}\} \to [0, 1]$
- additional conditions: has root node, terminal nodes terminal nodes $\mathcal{T}$ have no children, $p(\cdot)$ is properly defined (i.e. probabilities at chance node sum up to 1)

Decision tree

Goal: find a decision function
d, $d\colon \{e \in E : e_1 \in \mathcal{D}\} \to \{0,1\}$,
$\forall v \in \mathcal{D} : \sum_{e \in E, e_1 = v} d(e) = 1$

that maximizes the payoff

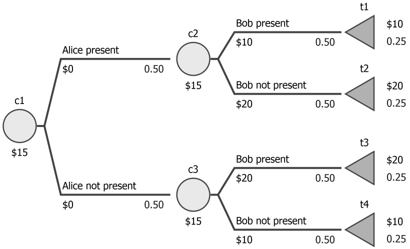
$$P(DT,d) = \begin{cases} 0 & \text{if} \quad r(DT) \in \mathcal{T} \\ \sum_{e:e_1=r(DT)} p(e)\left(y(e) + P\left(DT(e_2), d\right)\right) & \text{if} \quad r(DT) \in \mathcal{C} \\ \sum_{e:e_1=r(DT)} d(e)\left(y(e) + P\left(DT(e_2), d\right)\right) & \text{if} \quad r(DT) \in \mathcal{D} \end{cases}$$

where $r(DT)$ denotes the root of DT.

A strategy d maximizing $P(DT,d)$ is called *P-optimal*

Decision tree types

- Separable (changing the probabilities in one chance node does not automatically require changing any probabilities in any other chance node)
- Non-Separable (see the picture below)



Sensitivity analysis

1. $s \colon \mathcal{C} \to [0, 1]$
   function representing whether a given node should be subject to sensitivity analysis. $s(c) = 1$ denotes that c should be subject to sensitivity analysis
   $s(c) = 0$ denotes that probabilities are given precisely
   Values between 0 and 1 could denote various degrees of ambiguity
2. Stability analysis
   how stable a P-optimal strategy is
3. Perturbation analysis
   optimal strategy for various scenarios of probability perturbation

Sensitivity analysis - stability

- distance between two decision trees $DT' = ((V,E),p',y')$ and $DT = ((V,E),p,y)$

$$||DT, DT'||_s = \max_{e \in E, e_1 \in \mathcal{C}} \frac{|p'(e) - p(e)|}{s(e_1)}.$$

- The P-optimal strategy d is ε-stable, if it is also P-optimal for any DT' such that $||DT, DT'||_s \leq \varepsilon$
- stability index of a P-optimal strategy d:

$$I(DT, d, s) = \sup\{\varepsilon \in [0, +\infty] \colon d \text{ is } \varepsilon\text{-stable}\}$$

Theorem

Take a separable proper decision tree, DT, with a sensitivity function, $s(\cdot)$.

For any two P-optimal strategies, d_1 and d_2, we have $I(DT, d_1, s) = I(DT, d_2, s)$.

Sensitivity analysis - perturbation

- minimum positive sensitivity value
  $\widetilde{s} = \min\{s(v) : v \in \mathcal{C}, s(v) > 0\}$
- worst-case-tending expected payoff:

$$P_{\min}(DT, d, s, \varepsilon) = \min_{\{DT' : \, ||DT, DT'||_s \leq \varepsilon\}} P(DT', d).$$

- best-case-tending perturbation:

$$P_{\max}(DT, d, s, \varepsilon) = \max_{\{DT' : \, ||DT, DT'||_s \leq \varepsilon\}} P(DT', d)$$

Sensitivity analysis - perturbation cont.

- divergence function

$$D_{KL}(P||Q) = \sum_{i \in A} P(i) \log\left(\frac{P(i)}{Q(i)}\right),$$

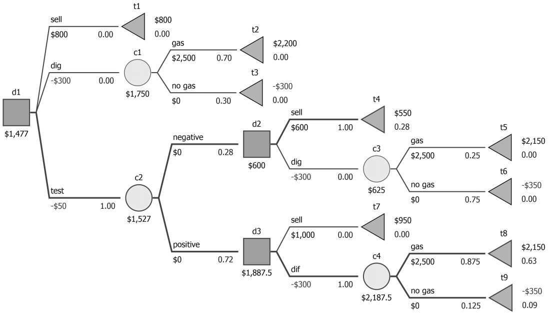
$$\begin{aligned} &\text{minimize} && D_{KL}(\mathbf{x}||\mathbf{p}) \\ &\text{subject to:} && D_{KL}(\mathbf{x}||\mathbf{u}) = \theta \wedge \mathbf{x} \geq (0) \wedge \sum_{i=1}^{n} x_i = 1, \end{aligned}$$

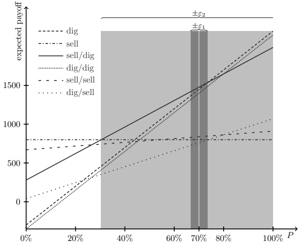
where $\mathbf{u} = (\frac{1}{n}, \ldots, \frac{1}{n})$ (of length n)

- The solution for the above optimization problem for various values of θ yields $\mathbf{x}$ changing along the path given by the following soft-max formula with parameter $\gamma \in [0, +\infty[$:

$$x_i = \frac{p_i^\gamma}{\sum_{j=1}^{n} p_j^\gamma}.$$

Non separable decision tree - sample





$$P(\text{pos. test}) = \text{sensitivity} \times P(\text{gas}) + (1 - \text{specificity}) \times P(\text{no gas}),$$

$$P(\text{neg. test}) = (1 - \text{sensitivity}) \times P(\text{gas}) + \text{specificity} \times P(\text{no gas}),$$

$$P(\text{gas}|\text{pos. test}) = \frac{\text{sensitivity} \times P(\text{gas})}{P(\text{pos. test})},$$

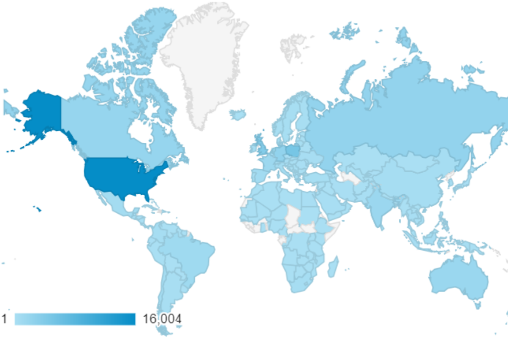
$$P(\text{gas}|\text{neg. test}) = \frac{(1 - \text{sensitivity}) \times P(\text{gas})}{P(\text{neg. test})},$$

Software

- SilverDecisions.pl (`http://silverdecisions.pl/`)
    - Graphical user interface, web based
    - Over 50'000 unique users (2'200 from Russia)
    - Javascript
    - Exports files to JSON
    - Fully open source - hosted on Git
- Chondro (`https://github.com/pszufe/chondro`)
    - A callable API
    - Python
    - Imports JSON decision tree definitions
        - Internal format (a nested Python dictionary representing a tree structure)
        - Trees exported from SilverDecision
    - Fully open source - hosted on Git

SilverDecisions.pl Users

	Country	Users	Sessions
		52,854 % of Total: 100.00% (52,854)	77,293 % of Total: 100.00% (77,293)
1.	United States	**16,004** (30.33%)	21,710 (28.09%)
2.	Poland	**4,969** (9.42%)	11,349 (14.68%)
3.	United Kingdom	**3,754** (7.12%)	4,723 (6.11%)
4.	Germany	**2,359** (4.47%)	3,175 (4.11%)
5.	Russia	**2,280** (4.32%)	3,077 (3.98%)
6.	Canada	**1,966** (3.73%)	2,515 (3.25%)
7.	Netherlands	**1,657** (3.14%)	2,466 (3.19%)
8.	Australia	**1,557** (2.95%)	2,289 (2.96%)
9.	India	**1,493** (2.83%)	1,896 (2.45%)
10.	France	**1,294** (2.45%)	1,630 (2.11%)



Chondro - typical usage scenario

1. Create a JSON representation of a DT (e.g. by saving a DT from SilverDecisions
2. Load the tree and use the `solve_tree` function to calculate optimal decision for the DT.
3. Perform the sensitivity analysis
   - use `find_stability` to calculate the stability
   - use `find_perturbation_mode` to calculate $P_{mode,\varepsilon}$
   - use `find_perturbation_pessopty` to calculate $P_{\min,\varepsilon}$ and $P_{\max,\varepsilon}$

```
d1:decision
  *c1:chance (ev=48)
    t1:p=3/5 final [60]
    t2:p=2/5 final [30]
   c2:chance (ev=44)
    t3:p=7/10 final [20]
    t4:p=3/10 final [100]
   c3:chance (ev=44)
    t5:p=4/5 final [40]
    t6:p=1/5 final [60]
```

Summary - results

1. Conceptual framework for sensitivity analysis of decision trees;
2. Methodology for performing SA when values in several nodes change simultaneously
3. Open Source software implementation that enables practical application of the concepts discussed in the paper
    - SilverDecisions.pl
        - currently over 50'000 users (most from the USA)
        - 2'200 users from Russia
    - Chondro (online Python library for tree stability analysis)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645860