Decision Tree

Theory, Rollback Method and Tool Description

English

Wolfgang Boehme

wolfgang.boehme@economic-modeling.de

www.economic-modeling.de

25 October 2026

Decision theory under risk – sequential decisions with expected values

The documentation on this website was developed with the support of the AI Claude (Anthropic).

Table of Contents

1 Introduction **2**

2 Key Concepts **2**

3 Rollback Method (Backward Induction) **2**

4 Complete Example **3**

4.1 Problem Statement 3

4.2 Rollback Calculation 3

4.3 Graphical Representation 4

4.4 Summary of Expected Values 5

5 Tool Description **5**

5.1 Building the Tree 5

5.2 Output 6

6 Summary **6**

1 Introduction

A **decision tree** models sequential decision problems in which multiple decisions and chance events alternate. It extends the simple decision matrix to *multi-stage* situations.

Typical Applications

- Investment decisions with multiple stages
- Product development with market launch risk
- Medical treatment decisions
- Corporate strategies with competitor responses

2 Key Concepts

Definition: Node Types

A decision tree consists of three node types:

- **Decision node** (rectangle, blue): The decision maker chooses a branch. Optimal choice: branch with the highest expected value.
- **Chance node** (circle, orange): Nature selects a branch according to probabilities $p_i \geq 0$, $\sum_i p_i = 1$.
- **Outcome node** (rectangle, green): Leaf node with payoff e.

Definition: Expected Value of a Node

The **expected value** (EV) is computed recursively from the leaves to the root (**Rollback method**):

$$\text{EV}(\text{Outcome}) = e \qquad \text{(Payoff)}$$

$$\text{EV}(\text{Chance node}) = \sum_i p_i \cdot \text{EV}(\text{Child}_i)$$

$$\text{EV}(\text{Decision node}) = \max_i \text{EV}(\text{Child}_i)$$

3 Rollback Method (Backward Induction)

Algorithm: Rollback

1. Start at the **Outcome nodes** (leaves) – EV = payoff.
2. Compute backwards to the root:
   - **Chance node:** $\text{EV} = \sum_i p_i \cdot \text{EV}_i$

- **Decision node:** $\text{EV} = \max_i \text{EV}_i$, non-optimal branches are *pruned*.

3. The EV of the root is the optimal expected value of the project.

Theorem: Optimality of Rollback

The Rollback method yields the **globally optimal strategy** – i.e. the decision rule that maximises the expected value across all possible future scenarios. This follows from the *Principle of Optimality* (Bellman, 1957).

4 Complete Example

4.1 Problem Statement

Example: Product Development (Two-Stage)

A company decides whether to develop a new product.

Stage 1 – Development decision:

- **Develop** (cost: 50): Leads to Chance node "Market test".
- **Do not develop**: Payoff 0.

Stage 2 – Market test (Chance node):

- **Positive** ($P = 0.6$): Second decision – Large series or Small series.
- **Negative** ($P = 0.4$): Project cancelled, payoff -50.

Stage 3 – Series decision (positive test):

- **Large series**: Chance node – Boom ($P = 0.7$, profit 300) or Flop ($P = 0.3$, loss -100).
- **Small series**: Chance node – Boom ($P = 0.7$, profit 120) or Flop ($P = 0.3$, profit 20).

All values in thousand euros, net of development costs.

4.2 Rollback Calculation

Step 1 – Outcome nodes: Large series-Boom: 300, Large series-Flop: -100, Small series-Boom: 120, Small series-Flop: 20, Negative: -50, Do not develop: 0.

Step 2 – Chance node Large series:

$$\text{EV}_{\text{Large}} = 0.7 \cdot 300 + 0.3 \cdot (-100) = 210 - 30 = \mathbf{180}$$

Step 3 – Chance node Small series:

$$\text{EV}_{\text{Small}} = 0.7 \cdot 120 + 0.3 \cdot 20 = 84 + 6 = \mathbf{90}$$

Step 4 – Decision node Series decision:

$$\text{EV}_{\text{Series}} = \max(180,\ 90) = \mathbf{180} \quad \Rightarrow \text{Choose Large series}$$

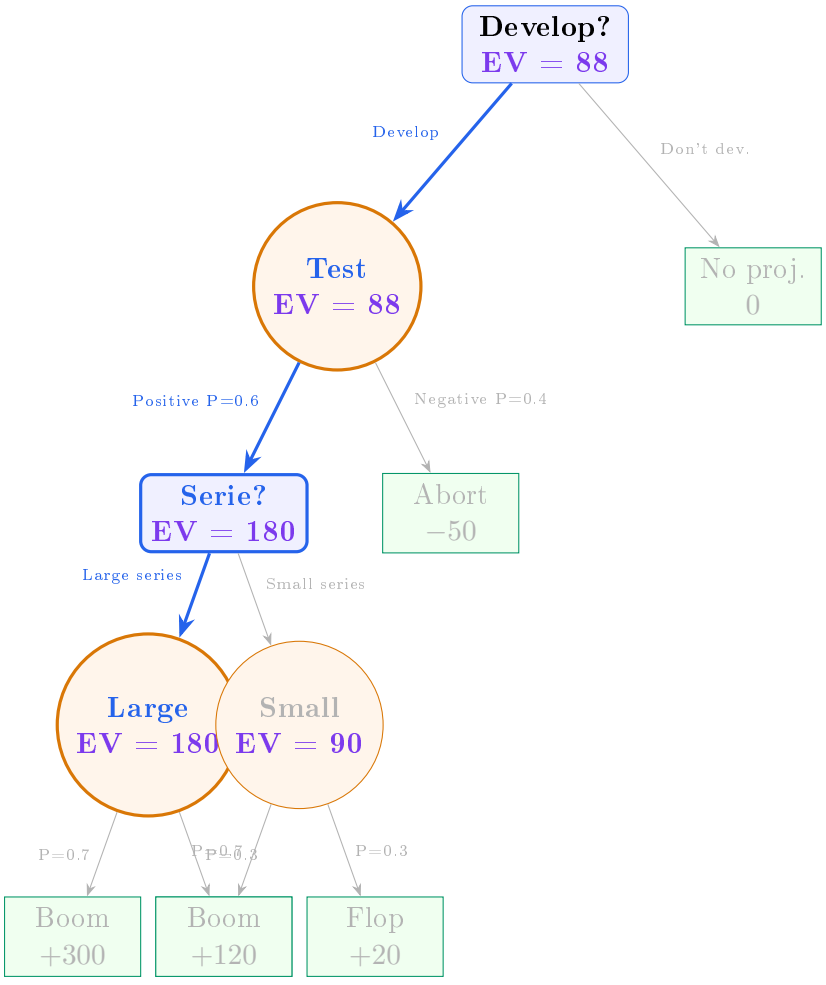
Step 5 – Chance node Market test:

$$\text{EV}_{\text{Test}} = 0.6 \cdot 180 + 0.4 \cdot (-50) = 108 - 20 = \mathbf{88}$$

Step 6 – Decision node (root):

$$\text{EV}_{\text{Root}} = \max(88,\ 0) = \mathbf{88} \quad \Rightarrow \text{Develop!}$$

4.3 Graphical Representation



Optimal strategy:

1. **Develop** the product (EV = 88 > 0).
2. If Market test **positive**: launch **Large series** (EV = 180 > 90).
3. If Market test negative: project cancelled.

4.4 Summary of Expected Values

Node	Type	EV	Optimal?
Root (Develop?)	Decision	88	Develop
Market test	Chance	88	–
Series decision	Decision	180	Large series
Large series	Chance	180	–
Small series	Chance	90	–

5 Tool Description

Decision Tree Tool

Interactive tool for building and computing decision trees.

5.1 Building the Tree

Method 1 – Manual:

1. Click + **New (empty)** – an empty root decision node appears.
2. Enter a label.
3. Add child nodes: +**D** (Decision), +**C** (Chance), +**O** (Outcome).
4. For Chance nodes: enter probability (sum = 1).
5. For Outcome nodes: enter payoff.
6. Click ▶ **Calculate & Draw**.

Method 2 – CSV Import:
Format (semicolon-separated):

Column	Meaning
`id`	Unique row ID (integer)
`type`	`dec` / `chance` / `leaf`
`label`	Node label
`prob`	Probability (only for `chance` children)
`payoff`	Payoff (only for `leaf`)
`parent`	Parent node ID (empty = root)

Example CSV for the above example:

```
id  typ     label            prob  payoff  parent
1   dec     Develop?
2   chance  Market test                    1
3   leaf    Do not develop         0       1
4   dec     Series decision                2
5   leaf    Negativ          0.4   -50     2
```

5.2 Output

- **Editor:** All nodes as a list with EV displayed after calculation.
- **SVG tree:** Optimal path highlighted in blue; chance nodes as circles, decision nodes as rectangles.
- **Validation:** Warning if probabilities of a chance node do not sum to 1.
- **CSV export:** Save the current tree.

6 Summary

Key Points: Decision Tree

1. Decision trees model **sequential decisions** under risk.
2. Three node types: **Decision** (selects best branch), **Chance** (weighted average), **Outcome** (payoff).
3. The **Rollback method** computes EV from leaves to root.
4. At each decision node the branch with **maximum EV** is chosen.
5. The result is the **globally optimal strategy** (Bellman principle).
6. The tool supports manual entry, CSV import/export and two examples.