Decision making in the presence of uncertainty

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Selection based on expected values

• Until now: The optimal action choice was the option that maximized the expected monetary value.
• But is the expected monetary value always the quantity we want to optimize?
Selection based on expected values

• Is the expected monetary value always the quantity we want to optimize?
• Answer: Yes, but only if we are risk-neutral.

• But what if we do not like the risk (we are risk-averse)?
• In that case we may want to get the premium for undertaking the risk (of loosing the money)
• Example:
  – we may prefer to get $101 for sure against $102 in expectation but with the risk of loosing the money
• Problem: How to model decisions and account for the risk?
• Solution: use utility function, and utility theory

Utility function

• Utility function (denoted U)
  – Quantifies how we “value” outcomes, i.e., it reflects our preferences
  – Can be also applied to “value” outcomes other than money and gains (e.g. utility of a patient being healthy, or ill)
• Decision making:
  – uses expected utilities (denoted EU)

  \[
  EU(X) = \sum_{x \in \Omega} P(X = x) U(X = x)
  \]

  \[
  U(X = x) \quad \text{the utility of outcome } x
  \]

Important !!
• Under some conditions on preferences we can always design the utility function that fits our preferences
Utility theory

- Defines axioms on preferences that involve uncertainty and ways to manipulate them.
- Uncertainty is modeled through lotteries
  - **Lottery:**
    \[ p : A ; (1 - p) : C \]
  - Outcome A with probability p
  - Outcome C with probability (1-p)
- The following six constraints are known as the axioms of utility theory. The axioms are the most obvious semantic constraints on preferences with lotteries.
- **Notation:**
  - \( \succ \) - preferable
  - \( \sim \) - indifferent (equally preferable)

Axioms of the utility theory

- **Orderability:** Given any two states, the a rational agent prefers one of them, else the two as equally preferable.
  \[(A \succ B) \lor (B \succ A) \lor (A \sim B)\]
- **Transitivity:** Given any three states, if an agent prefers A to B and prefers B to C, agent must prefer A to C.
  \[(A \succ B) \land (B \succ C) \Rightarrow (A \succ C)\]
- **Continuity:** If some state B is between A and C in preference, then there is a \( p \) for which the rational agent will be indifferent between state B and the lottery in which A comes with probability p, C with probability (1-p).
  \[(A \succ B \succ C) \Rightarrow \exists p \ [ p : A ; (1 - p) : C ] \sim B\]
Axioms of the utility theory

- **Substitutability**: If an agent is indifferent between two lotteries, $A$ and $B$, then there is a more complex lottery in which $A$ can be substituted with $B$.

  \( (A \sim B) \Rightarrow \[ p : A; (1 - p) : C \] \sim \[ p : B; (1 - p) : C \] \)

- **Monotonicity**: If an agent prefers $A$ to $B$, then the agent must prefer the lottery in which $A$ occurs with a higher probability.

  \( (A \succ B) \Rightarrow (p > q \Leftrightarrow \[ p : A; (1 - p) : B \] \succ \[ q : A; (1 - q) : B \]) \)

- **Decomposability**: Compound lotteries can be reduced to simpler lotteries using the laws of probability.

  \[
  \begin{align*}
  &\[ p : A; (1 - p) : [q : B; (1 - q) : C] \] \Rightarrow \\
  &\[ p : A; (1 - p)q : B; (1 - p)(1 - q) : C \]
  \end{align*}
  \]

Utility theory

**If the agent obeys the axioms of the utility theory, then**

1. There exists a real valued function $U$ such that:

   \[
   U(A) > U(B) \Leftrightarrow A \succ B \\
   U(A) = U(B) \Leftrightarrow A \sim B
   \]

2. The utility of the lottery is the expected utility, that is the sum of utilities of outcomes weighted by their probability.

   \[
   U[p : A; (1 - p) : B] = pU(A) + (1 - p)U(B)
   \]

3. Rational agent makes the decisions in the presence of uncertainty by maximizing its expected utility.
Utility functions

We can design a utility function that fits our preferences if they satisfy the axioms of utility theory.

- But how to design the utility function for monetary values so that they incorporate the risk?
- What is the relation between utility function and monetary values?
  - Assume we loose or gain $1000.
    - Typically this difference is more significant for lower values (around $100 - $1000) than for higher values (~ $1,000,000)
  - What is the relation between utilities and monetary value for a typical person?

Utility functions

- What is the relation between utilities and monetary value for a typical person?
- Concave function that flattens at higher monetary values
Utility functions

- Expected utility of a sure outcome of 750 is 750

\[ \text{EU(sure 750)} \]

Utility functions

Assume a lottery \( L \) [0.5: 500, 0.5:1000]

- Expected value of the lottery = 750
- Expected utility of the lottery \( \text{EU}(L) \) is different:
  - \( \text{EU}(L) = 0.5U(500) + 0.5*U(1000) \)
Utility functions

- Expected utility of the lottery $\text{EU}(\text{lottery } L) < \text{EU}(\text{sure } 750)$

- Risk aversion – a bonus is required for undertaking the risk

Decision making with utility function

- Original problem with monetary outcomes
Decision making with the utility function

- Utility function log (x)

![Decision Tree Diagram]

CS 1571 Introduction to AI
Lecture 26-b

Learning

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square
Machine Learning

- The field of **machine learning** studies the design of computer programs (agents) capable of learning from past experience or adapting to changes in the environment.

- The need for building agents capable of learning is everywhere:
  - Predictions in medicine, text classification, speech recognition, image/text retrieval, commercial software.

- Machine learning is not only the deduction but induction of rules from examples that facilitate prediction and decision making.

---

Learning

**Learning process:**
Learner (a computer program) takes data $D$ representing past experiences and tries to either:
- to develop an appropriate response to future data, or
- describe in some meaningful way the data seen.

**Example:**
Learner sees a set of past patient cases (patient records) with corresponding diagnoses. It can either try:
- to predict the presence of a disease for future patients
- describe the dependencies between diseases, symptoms (e.g. builds a Bayesian network for them).
Types of learning

• **Supervised learning**
  – Learning mapping between inputs $x$ and desired outputs $y$
  – Teacher gives me $y$’s for the learning purposes

• **Unsupervised learning**
  – Learning relations between data components
  – No specific outputs given by a teacher

• **Reinforcement learning**
  – Learning mapping between inputs $x$ and desired outputs $y$
  – Critic does not give me $y$’s but instead a signal (reinforcement) of how good my answer was

• **Other types of learning:**
  – Concept learning, explanation-based learning, etc.

---

Supervised learning

**Data:** $D = \{d_1, d_2, ..., d_n\}$ \textit{a set of $n$ examples}

$d_i = \langle x_i, y_i \rangle$

$x_i$ is input vector, and $y$ is desired output (given by a teacher)

**Objective:** learn the mapping $f : X \rightarrow Y$

s.t. \quad \quad y_i \approx f(x_i) \quad \text{for all} \quad i = 1, ..., n$

**Two types of problems:**

• **Regression:** $X$ discrete or continuous $\rightarrow$
  $Y$ is \textbf{continuous}

• **Classification:** $X$ discrete or continuous $\rightarrow$
  $Y$ is \textbf{discrete}
Supervised learning examples

- **Regression:** Y is *continuous*

<table>
<thead>
<tr>
<th>Debt/equity</th>
<th>Earnings</th>
<th>Future product orders</th>
<th>company stock price</th>
</tr>
</thead>
</table>

- **Classification:** Y is *discrete*

```
7 7 7 7
7 7 7 7
7 7 7 7
```

Label “3”

Handwritten digit (array of 0,1s)

Unsupervised learning

- **Data:** \( D = \{d_1, d_2, \ldots, d_n\} \)
  \[ d_i = \mathbf{x}_i \quad \text{vector of values} \]
  No target value (output) y

- **Objective:**
  - learn relations between samples, components of samples

Types of problems:

- **Clustering**
  Group together “similar” examples, e.g. patient cases
- **Density estimation**
  - Model probabilistically the population of samples, e.g. relations between the diseases, symptoms, lab tests etc.
Unsupervised learning example.

- **Density estimation.** We want to build the probability model of a population from which we draw samples \( d_i = x_i \)

Unsupervised learning. Density estimation

- A probability density of a point in the two dimensional space
  - Model used here: Mixture of Gaussians
Reinforcement learning

- We want to learn: $f : X \rightarrow Y$
- We see samples of $x$ but not $y$
- Instead of $y$ we get a feedback (reinforcement) from a critic about how good our output was

- The goal is to select output that leads to the best reinforcement

Typical learning

Assume we have an access to the dataset $D$ (past data)

Three basic steps:
- Select a model with parameters
- Select the error function to be optimized
  - Reflects the goodness of fit of the model to the data
- Find the set of parameters optimizing the error function
  - The model and parameters with the smallest error represent the best fit of the model to the data
Learning

• Assume we see examples of pairs \((x, y)\) and we want to learn the mapping \(f : X \rightarrow Y\) to predict future \(y\)s for values of \(x\)

![Graph showing data points and a line of best fit]

Learning bias

• **Problem:** many possible functions \(f : X \rightarrow Y\) exists for representing the mapping between \(x\) and \(y\)
• We choose a class of functions. Say we choose a linear function: \(f(x) = ax + b\)

![Graph showing data points and a linear trend line]
Learning

• Choosing a parametric model or a set of models is not enough
  Still too many functions $f(x) = ax + b$
  – One for every pair of parameters $a, b$

\[
\sum_{i=1}^{n} (y_i - f(x_i))^2
\]

Learning

• Optimize the model using some criteria that reflects the fit of the
  model to data
• Example: mean squared error $\frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$
Typical learning

Assume we have an access to the dataset D (past data)

Three basic steps:

- **Select a model** with parameters
  \[ f(x) = ax + b \]

- **Select the error function** to be optimized
  – Reflects the goodness of fit of the model to the data
  \[
  \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2
  \]

- **Find the set of parameters optimizing the error function**
  – The model and parameters with the smallest error represent the best fit of the model to the data