Random Variables

Def: A Random Variable is defined by a function that assigns a value to every element of the sampling space $\Omega$.

For example: launching two fair coins: head (T) or tail (X)
- TTT elements of $\Omega$
- XTT

Random variable $t$: number of $T$ in every triple of launches
- $[TTT \rightarrow 3]$ The values "2" and "3" are the associated values.
- $[XTT \rightarrow 2]$
Random Variables

Def. Random Variable (formal definition)
A Random Variable is a measurable function

\[ X : \Omega \to \mathbb{R} \]

where \((\mathbb{R}, B(\mathbb{R}))\) is a Borel space on the set of real numbers.

Properties

Given a probability space \((\Omega_1, \mathcal{F}_1, \mathbb{P})\), measurable space \((\Omega_2, \mathcal{F}_2)\) and measurable function \(f : \Omega_1 \to \Omega_2\) then

\[(\Omega_2, \mathcal{F}_2, \mathbb{P} \circ f^{\text{orig}})\]

is a probability space

where \(\circ\) stands for function composition: \((f \circ g)x = f(g(x))\)

Example

Random variable \(t\) : number of T in a triple launch of a coin
Random Variables

\[ P\{t \geq 2\} \\
= \{\text{Shorthand for}\} \\
P\{\ell \in \Omega_1 | t \ell \geq 2\} \\
= \{t \ell \geq 2 \equiv t \ell = 2 \lor t \ell = 3\} \\
P\{\ell \in \Omega_1 | t \ell = 2 \lor t \ell = 3\} \\
= (P \circ t_{\text{orig}}\{2, 3\} \\
= P\{\text{TXT, XTT, TTX, TTT}\} \\
= \frac{4}{8} \\
= \frac{1}{2} \]

Random Variables

\[ P\{t = 3\} \]

\[ P\{\ell \in \Omega_1 | t \ell = 3\} = (P \circ t_{\text{orig}})3 = P\{TTT\} = \frac{1}{8} \]

Discrete Random Variables: assign discrete values

Continuous Random Variables: assign continuous values
Stochastic Processes
Def: A stochastic process is a $\mathcal{T}$-indexed family of random variables:
$$\{X_i | i \in \mathcal{T}\}$$

Def: Discrete Stochastic Process
$$\mathcal{T} = \mathbb{N} : \{X_n | n \in \mathbb{N}\}$$
Note that every $X_i$ in turn may be a discrete or continuous random variable.

Examples:
- Discrete stochastic process with a discrete random variable
  - Number of emails one receives every day
  - Finite state system: every step gives the state in which the system is
- Discrete stochastic process with a continuous random variable:
  - Mean daily temperature

Stochastic Processes
Def: Continuous Stochastic Process
$$\mathcal{T} = \mathbb{R} : \{X_t | t \in \mathbb{R}\}$$
Also in this case every $X_t$ in turn may be a discrete or continuous random variable.

Examples:
- Continuous stochastic process with a discrete random variable
  - Number of people on a bus at any time
  - Number of wireless network nodes active at any time
- Continuous stochastic process with a continuous random variable:
  - Temperature at every time of the day
Stochastic Processes

Stochastic Processes that describe (system) behaviour

\[
P\{\omega \in \Omega|X_0 \omega = \text{Ready}\} = 1
\]

\(X_0\) associates to a computation the state at step 0

\[
P\{X_0 = \text{Ready}\} = 1
P\{X_0 = \text{Run}\} = 0
P\{X_0 = \text{Wait}\} = 0
\]

Starting from the initial state “Ready”

\[
P\{X_1 = \text{Run}|X_0 = \text{Ready}\} = ?
\] and
\[
P\{X_2 = \text{Wait}|X_0 = \text{Ready}\} = ?
\]

Discrete Time Markov Chain (DTMC)

Discrete Time Markov Chain

A Discrete Time Markov Chain is a special case of a Stochastic Process with discrete steps and a discrete space

\[
\{X_n|n \in \mathbb{N}\}
\]

such that it satisfies the following two properties:

- Markov property
- Time homogeneity
Discrete Time Markov Chain (DTMC)

- **Markov property**

\[
P\{X_{n+1} = i_{n+1}|X_0 = i_0, \ldots X_n = i_n\} = P\{X_{n+1} = i_{n+1}|X_n = i_n\}
\]

The probability at step \(n + 1\) depends only on the value of \(X_n\), so on that of the previous step.

- **Time homogeneity**

\[
P\{X_{n+1} = j|X_n = i\} = P\{X_{m+1} = j|X_m = i\}
\]

for all \(n, m \in \mathbb{N}\)

The probability to go from state \(i\) to state \(j\) does not depend on how many steps have been made to end up in \(i\).

For example, to obtain 6 by throwing a dice does not depend on the number of times the dice has been used in the past.

---

**Model Checking DTMC**

- **Plus:**
  - Temporal logic suitable to specify probability measures
  - User does not need to know all algorithmic details
  - "High-level debugger"
  - Both qualitative and quantitative aspects can be analysed
  - Separation of requirements from system model
  - Formal semantics

- **Minus:**
  - Combinatorial explosion of state space
  - No counter-examples generated
  - Modelling cannot be automated
  - "None of the errors found" (Dijkstra)
Model Checking DTMC

Labelled DTMC

A labelled DTMC $D$ is a triple $(S, P, L)$ where:

- $S$ is a finite set of discrete states
- $P : S \times S \to [0, 1]$ is a stochastic matrix i.e. $\sum_{s' \in S} P(s, s') = 1$
  - $P(s, s')$ is the discrete probability to move from state $s$ to state $s'$ in one step
  - a state is called absorbing if $P(s, s') = 0$ for all $s' \neq s \in S$
- $L : S \to 2^{AP}$ a labelling function
  - $L(s)$ is the set of atomic propositions that hold in $s$

Elementary probabilities of DTMC

Transient probability

$$\pi_j(n) = \mathbb{P}\{X_n = j\}$$

The probability that the DTMC $D$ is in state $j$ at step $n$.

$$\bar{\pi}(n) = (\pi_1(n), \ldots, \pi_{|S|}(n))$$

where $S$ is a set of states and $\bar{\pi}(n)$ a probability vector.

Iterative computation: given $\bar{\pi}(0)$ we have that $\bar{\pi}(n) = \bar{\pi}(0).P^n$ given $\bar{\pi}(0)$

- $\bar{\pi}(0)$: initial probability distribution
- $P$: stochastic matrix

Example: Parrow’s protocol

Assumptions:

- Ack does not get lost
- Probability of timeout: $\frac{1}{4}$
- Probability of ‘out’: $\frac{3}{4}$
Model Checking DTMC

- DTMC as a graph
- DTMC as a stochastic matrix

**Steady state probability**

\[ \pi = (\pi_0, \pi_1, \ldots, \pi_{|S|}) \]

\( \pi_i \): the probability to be in state \( s_i \) in the long term.

\[ \pi = \pi . P, \quad \sum_{i=1}^{|S|} \pi_i = 1, \quad 0 \leq \pi_i \leq 1 \]

if the limit exists.

In that case it is independent from \( \pi(0) \).

Example: Parrow’s protocol

\[ (\pi_1, \ldots, \pi_6) . P = (\pi_1, \ldots, \pi_6) \]

Solve the equations and use normalisation.
Model Checking DTMC

Example: Parrow’s protocol

\[(\pi_1, \ldots, \pi_6).P = (\pi_1, \ldots, \pi_6)\]

Solve the equations and use normalisation.

\[
\pi_1 = \frac{3}{7} \quad \pi_3 = \frac{1}{17}
\]

\[
\pi_2 = \frac{4}{17} \quad \pi_4 = \pi_5 = \pi_6 = \frac{3}{17}
\]

Probabilistic Computation Tree Logic (PCTL)

Probabilistic extension of the temporal logic CTL.

Syntax of PCTL:
Given AP – a set of Atomic Propositions \( a \in AP \)

PCTL formulas are defined as follows:

\[ F ::= \Phi \quad \text{PCTL formulas are the state formulas} \]

State formulas:

\[ \Phi ::= a \quad \text{atomic proposition } a \in AP \]

\[ | \Phi \lor \Phi \quad \text{disjunction} \]

\[ | \neg \Phi \quad \text{negation} \]

\[ | P_{\leq p}(\phi) \quad \text{with probability } p, \leq \in \{<, \leq, >, \geq\}, \text{the probability of the set of paths that satisfy } \phi \text{ is } \leq p \]

Path formulas:

\[ \phi ::= X \Phi \quad \text{in the next step formula } \Phi \text{ holds} \]

\[ | \Phi U^{\leq k} \Psi \quad \text{where } k \in \mathbb{N}. \Phi U^{\leq k} \Psi: \Phi \text{ holds along the path until } \Psi \text{ holds which must happen within } k \text{ steps} \]

\[ | \Phi U \Psi \quad \Phi U \Psi: \text{eventually } \Psi \text{ holds along the path where, until then, } \Phi \text{ must hold} \]
Probabilistic Computation Tree Logic (PCTL)

PCTL semantics:
We define the satisfaction relation:

\[ \mathcal{D}, s \models \Phi \]

"In the structure \( \mathcal{D} = (S, P, L) \), state \( s \) satisfies formula \( \Phi \)"

Semantics of PCTL path formulas:

We define the satisfaction relation:

\[ \mathcal{D}, \sigma \models_{\text{pt}} \phi \]

"In the structure \( \mathcal{D} = (S, P, L) \), path \( \sigma \) satisfies path formula \( \phi \)"

\[ \mathcal{D}, \sigma \models_{\text{pt}} X \phi \quad \text{iff} \quad \mathcal{D}, \sigma[1] \models_{\text{st}} \phi \]

\[ \mathcal{D}, \sigma \models_{\text{pt}} \Phi U^k \psi \quad \text{iff} \quad \exists 0 \leq i \leq k, \sigma[i] \models_{\text{st}} \psi \quad \text{and} \quad (\forall 0 \leq i < j, \sigma[i] \models_{\text{st}} \Phi) \]

\[ \mathcal{D}, \sigma \models_{\text{pt}} \Phi U \psi \quad \text{iff} \quad \exists k \geq 0, \mathcal{D}, \sigma \models_{\text{pt}} \Phi U^k \psi \]

Example for Parrow’s protocol

- "The probability that a time out happens in the next step is less than 0.3"
Probabilistic Computation Tree Logic (PCTL)

Example for Parrow’s protocol

• “The probability that a time out happens in the next step is less than 0.3”
  \[ P_{\leq 0.3}(X \text{ timeout}) \]

• “The probability that a time out occurs within the next 5 steps is less than 0.5”
  \[ P_{\leq 0.5}(tt U \leq 5) = P_{\leq 0.5}(\Diamond \leq 5 \Phi) \]

Probabilistic Computation Tree Logic (PCTL)

Example for Parrow’s protocol

• “The probability that a time out happens in the next step is less than 0.3”
  \[ P_{\leq 0.3}(X \text{ timeout}) \]

• “The probability that a time out occurs within the next 5 steps is less than 0.5”

Some shorthands: eventually and always

\[ P_{\leq p}(\Diamond \leq k \Phi) = P_{\leq p}(tt U \leq k \Phi) \]

and

\[ P_{\geq p}(\Box \leq k \Phi) = P_{\leq 1-p}(\Diamond \leq k \neg \Phi) \]