Modeling and Verification of Probabilistic Systems

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

http://moves.rwth-aachen.de/teaching/ws-1819/movep18/

October 08, 2018

Overview

Theme of the course

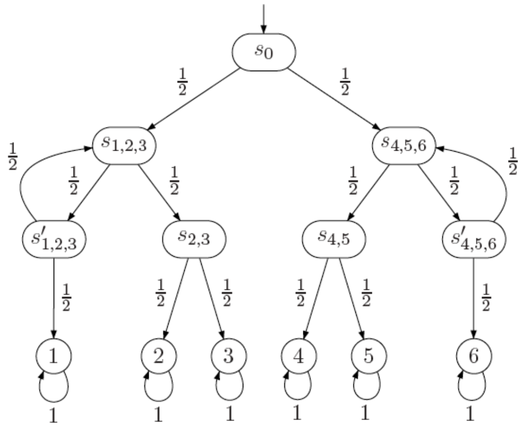
The theory of modelling and verification of probabilistic systems

Overview

More than five reasons for probabilities

1. Randomised Algorithms
2. Reducing Complexity
3. Probabilistic Programming
4. Reliability
5. Performance
6. Optimisation
7. Systems Biology

Randomised algorithms: Simulating a die [Knuth & Yao, 1976]



Heads = "go left"; tails = "go right". Does this model a six-sided die?

Distributed computing

FLP impossibility result **[Fischer *et al.*, 1985]**

In an asynchronous setting, where only one processor might crash, there is no distributed algorithm that solves the consensus problem—getting a distributed network of processors to agree on a common value.

Ben-Or's possibility result **[Ben-Or, 1983]**

If a process can make a decision based on its internal state, the message state, and some probabilistic state, consensus in an asynchronous setting is almost surely possible.

Example: Self-stabilisation

A distributed algorithm is self-stabilising iff:

- Convergence:
  Starting from an arbitrary state, it will always converge to a legitimate state.
- Closure:
  And it remains in a legitimate set of states thereafter in absence of faults.

A self-stabilising algorithm:

- Works correctly for every initialisation
- Recovers from the occurrence of transient faults

A key concept in fault-tolerant distributed computing

Dijkstra's Self-Stabilising Algorithm

- Asynchronous processes $0, \ldots, N$ form a directed ring
- Process i has a variable $x_i \in \{0, \ldots, K-1\}$, for $K \geqslant N$
- Processes have access to their neighbour's variables, and execute:
  - Process 0: if $x_0 = x_N$, then $x_0 := (x_0+1) \bmod K$
  - Process $i \neq 0$: if $x_i \neq x_{i-1}$ then $x_i := x_{i-1}$
- Process with enabled guard holds a token
- Legitimate state = unique token

Performance metric = worst-case convergence time

Symmetric Self-Stabilisation

Dijkstra's algorithm uses a designated process to break the symmetry

Self-stabilisation in anonymous networks is impossible

Possible solution: use randomisation.

Randomised Self-Stabilisation

A distributed randomised algorithm is stabilising iff:

- Convergence:
  Starting from an arbitrary state, it will almost surely converge to a legitimate state
- Closure:
  And it remains in a legitimate set of states thereafter in absence of faults

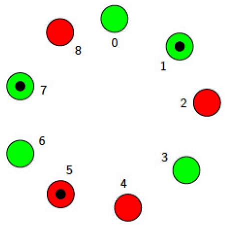
Herman's algorithm is a prime example of such algorithm

Herman's Randomised Self-Stabilisation

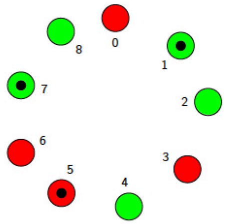
- $N+1$ (odd) synchronous processes $0, \ldots, N$ form a directed ring
- Process i has a Boolean variable $x_i \in \{0, 1\}$
- Processes have access to their neighbour's variables
- Process i performs:
  - if $x_i = x_{i-1}$, then $x_i := \begin{cases} 0 & \text{with probability } 1/2 \\ 1 & \text{with probability } 1/2 \end{cases}$
  - if $x_i \neq x_{i-1}$ then $x_i := x_{i-1}$
- Process has token if x_i equals x_{i-1}

Performance metric = expected convergence time

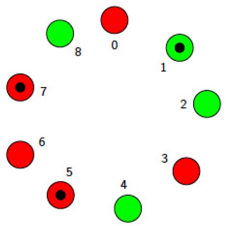
A Round of Herman's Algorithm



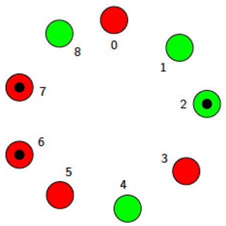
current configuration



toggle all processes w/o token

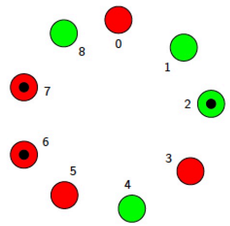


flip coins for every token

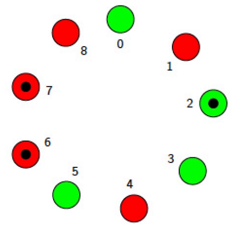


"assign" the tokens

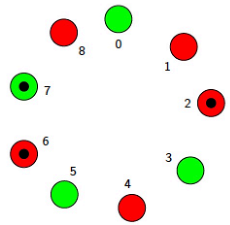
A Next Round



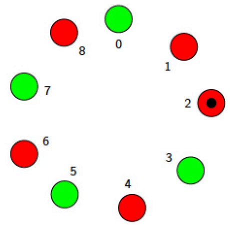
current configuration



toggle all processes w/o token



flip coins for every token



"assign" the tokens

Herman's Randomised Self-Stabilisation

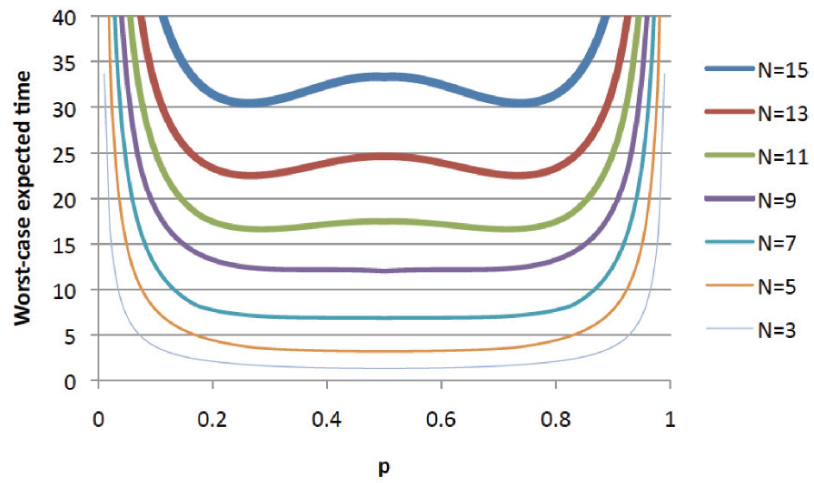
What is Herman's algorithm expected convergence time?

Consider Herman's original algorithm:

- Process i performs:
  - if $x_i = x_{i-1}$, then $x_i := \begin{cases} 0 & \text{with probability } p \\ 1 & \text{with probability } 1-p \end{cases}$
  - if $x_i \neq x_{i-1}$ then $x_i := x_{i-1}$
- Process hat token if x_i equals x_{i-1}

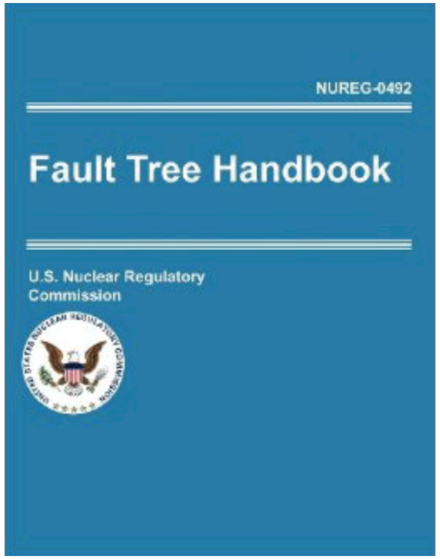
Use Biased Coins

[Kwiatkowska *et al.*, 2012]

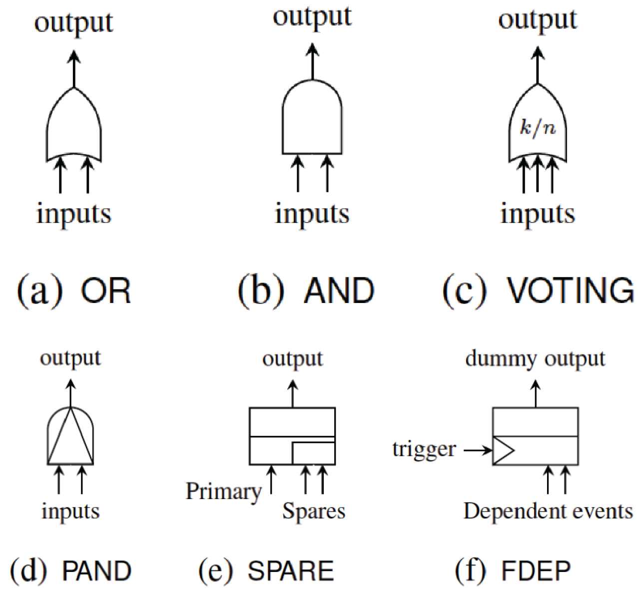


For larger rings, a biased coin reduces the expected convergence time

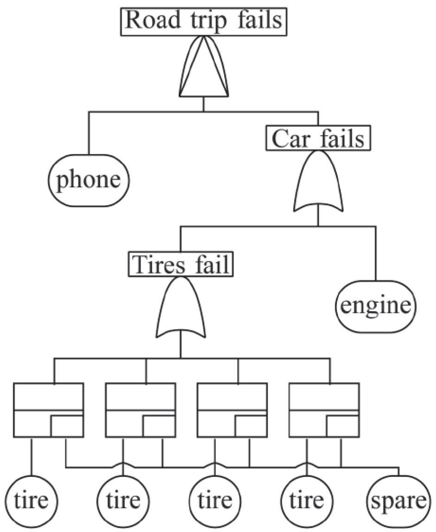
Reliability engineering



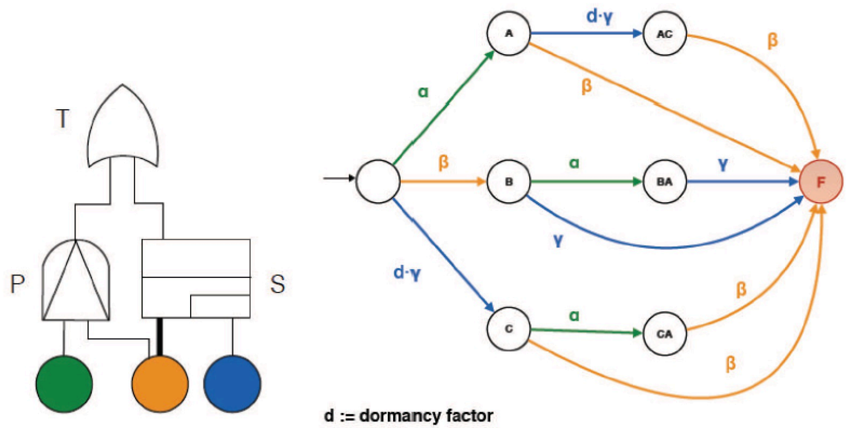
Reliability: (Dynamic) Fault Trees [Dugan *et al.*, 1990]



A fault tree example



Fault trees are Markov models

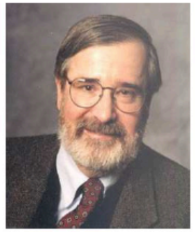


Probabilities help

- When modelling and analysing dependability and reliability
  - to quantify arrivals, message loss, waiting times, time between failure, QoS, ...
- When building protocols for networked embedded systems
  - randomized algorithms
- When problems are undecidable
  - repeated reachability of lossy channel systems, ...
- For obtaining a better performance
  - Freivald's matrix-mulitplication, random Quicksort ...

Topic of this lecture series

"Probabilistic model checking is one of the main challenges for the future."



Edmund J. Clarke

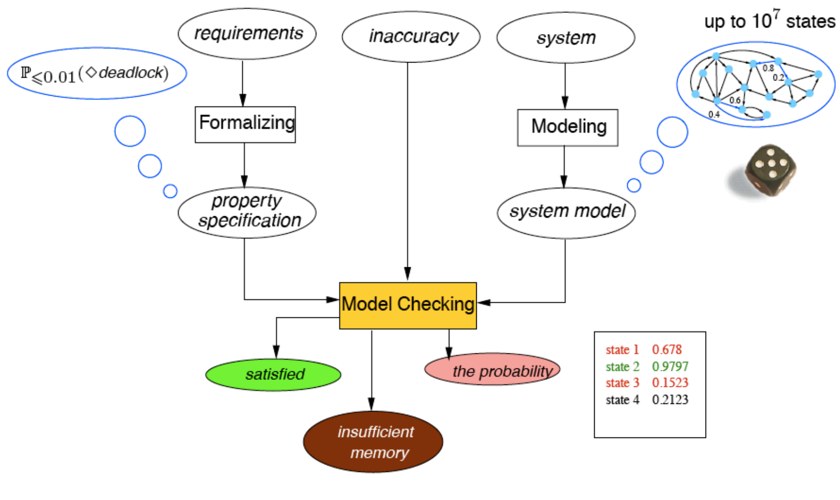
The Birth of Model Checking, 2008

Topic of this lecture series

"A promising new direction in formal methods research these days is the development of probabilistic models, with associated tools for quantitative evaluation of system performance along with correctness."

ACM SIGLOG News 2015

What is probabilistic model checking?



Probabilistic models

	Nondeterminism no	Nondeterminism yes
Discrete time	discrete-time Markov chain (DTMC)	Markov decision process (MDP)
Continuous time	CTMC	CTMDP

Some other models: probabilistic variants of (priced) timed automata

Properties

	Logic	Monitors
Discrete time	probabilistic CTL	deterministic automata (safety and LTL)
Continuous time	probabilistic timed CTL	deterministic timed automata

Core problem: computing (timed) reachability probabilities

Overview

Course topics

A probability theory refrehser

- measurable spaces, σ-algebra, measurable functions
- geometric, exponential and binomial distributions
- Markov and memoryless property
- limiting and stationary distributions

What are probabilistic models?

- discrete-time Markov chains
- continuous-time Markov chains
- extensions of these models with rewards
- Markov decision processes (or: probabilistic automata)
- Markov automata

Course topics

What are properties?

- reachability probabilities, i.e., $\Diamond G$
- long-run properties
- linear temporal logic
- probabilistic computation tree logic

How to check temporal logic properties?

- graph analysis, solving systems of linear equations
- deterministic Rabin automata, product construction
- linear programming, integral equations
- uniformization, Volterra integral equations

Course topics

How to make probabilistic models smaller?

- Equivalences and pre-orders
- Which properties are preserved?

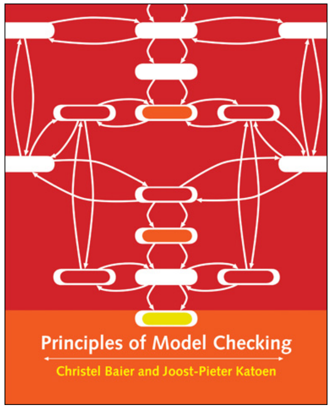
How to model probabilistic models?

- parallel composition and hiding
- compositional modelling and minimisation

Advanced topics

- multi-objective verification
- parameter synthesis

Course material



Ch. 10, Principles of Model Checking

Christel Baier

TU Dresden, Germany

Joost-Pieter Katoen

RWTH Aachen University, Germany, and University of Twente, the Netherlands

Other literature

- H.C. Tijms: A First Course in Stochastic Models. Wiley, 2003.
- H. Hermanns: Interactive Markov Chains: The Quest for Quantified Quality. LNCS 2428, Springer-Verlag, 2002.
- J.-P. Katoen. The Probabilistic Model Checking Landscape, LICS, 2016. (see course web page for download)
- J.-P. Katoen. Model Checking Meets Probability: A Gentle Introduction. IOS Press, 2013. (see course web-page for download)
- M. Stoelinga. Introduction to Probabilistic Automata. Bull. ETACS, 2002.
- M. Kwiatkowska *et al.*. Stochastic Model Checking. LNCS 4486, Springer-Verlag, 2007.

Lectures

Lecture

- Mon 10:30–12:00 (5056), Tue 08:30–10:00 (5056)
- Oct 8, 9, 15, 22, 23, 29, 30
- Nov 5, 6, 12, 13, 19, 20, 26, 27
- Dec 3, 10, 11, 17, 18
- January 7, 8 ......
- Check regularly course web page for possible "no shows"

Material

- Lecture slides (with gaps) are made available on web page
- Copies of the books are available in the CS library

Website

http://moves.rwth-aachen.de/teaching/ws-1819/movep18/

Exercises and exam

Exercise classes

- Wed 14:30 - 16:00 in AH 6 (start: Oct 24)
- Instructors: Tim Quatmann and Jip Spel

Weekly exercise series

- Intended for groups of 2 students
- New series: every Wed on course web page (start: Oct 24)
- Solutions: Wed (before 14:15) one week later

Exam:

- unknown date (written or oral exam)
- participation if $\geqslant$ 40% of all exercise points are gathered

Course embedding

Aim of the course

It's about the foundations of verifying and modelling probabilistic systems

Prerequisites

- Automata and language theory
- Algorithms and data structures
- Probability theory
- Introduction to model checking

Some related courses

- Stochastic Games (Löding)
- Probabilistic Programming (Katoen)

Questions?

Overview

Probability theory is simple, isn't it?

In no other branch of mathematics
is it so easy to make mistakes
as in probability theory

Henk Tijms, "Understanding Probability" (2004)

Measurable space

Sample space

A *sample space* Ω of a chance experiment is a set of elements that have a 1-to-1 relationship to the possible outcomes of the experiment.

σ-algebra

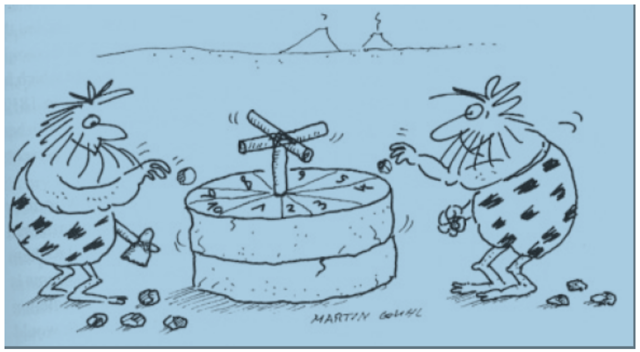
A *σ-algebra* is a pair $(\Omega, \mathcal{F})$ with $\Omega \neq \varnothing$ and $\mathcal{F} \subseteq 2^\Omega$ a collection of subsets of sample space Ω such that:

1. $\Omega \in \mathcal{F}$
2. $A \in \mathcal{F} \;\Rightarrow\; \Omega - A \in \mathcal{F}$ — complement
3. $(\forall i \geqslant 0.\ A_i \in \mathcal{F}) \;\Rightarrow\; \bigcup_{i \geqslant 0} A_i \in \mathcal{F}$ — countable union

The elements in $\mathcal{F}$ of a σ-algebra $(\Omega, \mathcal{F})$ are called *events*.
The pair $(\Omega, \mathcal{F})$ is called a *measurable space*.

Let Ω be a set. $\mathcal{F} = \{\varnothing, \Omega\}$ yields the smallest σ-algebra; $\mathcal{F} = 2^\Omega$ yields the largest one.

Probabilities



Probability space

Probability space

A *probability space* $\mathcal{P}$ is a structure $(\Omega, \mathcal{F}, Pr)$ with:

- $(\Omega, \mathcal{F})$ is a σ-algebra, and
- $Pr : \mathcal{F} \to [0,1]$ is a *probability measure*, i.e.:
  1. $Pr(\Omega) = 1$, i.e., Ω is the certain event
  2. $Pr\left(\bigcup_{i \in I} A_i\right) = \sum_{i \in I} Pr(A_i)$ for any $A_i \in \mathcal{F}$ with $A_i \cap A_j = \varnothing$ for $i \neq j$, where $\{ A_i \}_{i \in I}$ is finite or countably infinite.

The elements in $\mathcal{F}$ of a probability space $(\Omega, \mathcal{F}, Pr)$ are called *measurable* events.

Some lemmas

Properties of probabilities

For measurable events A, B and A_i and probability measure Pr:

- $Pr(A) = 1 - Pr(\Omega - A)$
- $Pr(A \cup B) = Pr(A) + Pr(B) - Pr(A \cap B)$
- $Pr(A \cap B) = Pr(A \mid B) \cdot Pr(B)$
- $A \subseteq B$ implies $Pr(A) \leqslant Pr(B)$
- $Pr(\bigcup_{n \geqslant 1} A_n) = \sum_{n \geqslant 1} Pr(A_n)$ provided A_n are pairwise disjoint

Discrete probability space

Discrete probability space

Pr is a *discrete* probability measure on $(\Omega, \mathcal{F})$ if

- there is a countable set $A \subseteq \Omega$ such that for $a \in A$:

$$\{ a \} \in \mathcal{F} \quad \text{and} \quad \sum_{a \in A} Pr(\{ a \}) = 1$$

- e.g., a probability measure on $(\Omega, 2^{\Omega})$

$(\Omega, \mathcal{F}, Pr)$ is then called a *discrete* probability space; otherwise, it is a *continuous probability* space.

Example

Example discrete probability space: throwing a die, number of customers in a shop, ...

Example

Example continuous probability space: throwing a dart on a circular board (see

Random variable

Measurable function

Let $(\Omega, \mathcal{F})$ and $(\Omega', \mathcal{F}')$ be measurable spaces. Function $f : \Omega \to \Omega'$ is a *measurable function* if

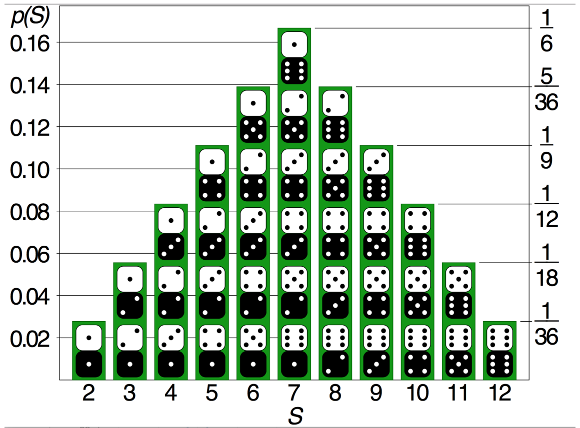
$$f^{-1}(A) = \{ a \mid f(a) \in A \} \in \mathcal{F} \quad \text{for all } A \in \mathcal{F}'$$

Random variable

Measurable function $X : \Omega \to \mathbb{R}$ is a *random variable*.

The *probability distribution* of X is $Pr_X = Pr \circ X^{-1}$ where Pr is a probability measure on $(\Omega, \mathcal{F})$.

Example: rolling a pair of fair dice



Distribution function

Distribution function

The *distribution function* F_X of random variable X is defined by:

$$F_X(d) \;=\; Pr_X((-\infty, d]) = Pr(\underbrace{\{ a \in \Omega \mid X(a) \leqslant d \}}_{\{ X \leqslant d \}}) \quad \text{for real } d$$

Properties

- F_X is monotonic and right-continuous
- $0 \leqslant F_X(d) \leqslant 1$
- $\lim_{d \to -\infty} F_X(d) = 0$ and
- $\lim_{d \to \infty} F_X(d) = 1$.

Discrete / continuous random variables

Distribution function

The *distribution function* F_X of random variable X is defined for $d \in \mathbb{R}$ by:

$$F_X(d) \;=\; Pr_X(X \in (-\infty, d]) = Pr(\{ a \in \Omega \mid X(a) \leqslant d \})$$

In the continuous case, F_X is called the *cumulative density function*.

Distribution function

- For discrete random variable X, F_X can be written as:

$$F_X(d) = \sum_{d_i \leqslant d} Pr_X(X{=}d_i)$$

- For continuous random variable X, F_X can be written as:

$$F_X(d) = \int_{-\infty}^{d} f_X(u)\; du \quad \text{with } f \text{ the density function}$$

Expectation and variance

Expectation

The *expectation* of discrete r.v. X with range I is defined by

$$E[X] \;=\; \sum_{x_i \in I} x_i \cdot Pr_X(X{=}x_i)$$

provided that this series converges absolutely, i.e., the sum must remain finite on replacing all x_i's with their absolute values.

The expectation is the weighted average of all possible values that X can take on.

Variance

The *variance* of discrete r.v. X is given by $Var[X] = E[X^2] - (E[X])^2$.

Stochastic process

Stochastic process

A *stochastic process* is a collection of random variables $\{ X_t \mid t \in T \}$.

- casual notation $X(t)$ instead of X_t
- with all X_t defined on probability space $\mathcal{P}$
- parameter t (mostly interpreted as "time") takes values in the set T

X_t is a random variable whose values are called *states*. The set of all possible values of X_t is the *state space* of the stochastic process.

State space	Parameter space T: Discrete	Parameter space T: Continuous
Discrete	# jobs at k-th job departure	# jobs at time t
Continuous	waiting time of k-th job	total service time at time t

Example stochastic processes

- Waiting times of customers in a shop
- Interarrival times of jobs at a production lines
- Service times of a sequence of jobs
- Files sizes that are downloaded via the Internet
- Number of occupied channels in a wireless network
- ......

Bernouilli process

Bernouilli random variable

Random variable X on state space $\{ 0, 1 \}$ defined by:

$$Pr(X = 1) = p \quad \text{and} \quad Pr(X = 0) = 1-p$$

is a *Bernouilli* random variable.

The mass function is given by $f(k; p) = p^k \cdot (1-p)^{1-k}$ for $k \in \{ 0, 1 \}$.

Expectation $E[X] = p$; variance $Var[X] = E[X^2] - (E[X])^2 = p \cdot (1-p)$.

Bernouilli process

A *Bernouilli process* is a sequence of independent and identically distributed Bernouilli random variables $X_1, X_2, \ldots$.

Binomial process

Binomial process

Let $X_1, X_2, \ldots$ be a Bernouilli process. The *binomial* process S_n is defined by $S_0 = 0$ and $S_n = \sum_{i=1}^{n} X_i$. The probability distribution of "counting process" S_n is given by:

$$Pr\{ S_n = k \} = \binom{n}{k} p^k \cdot (1-p)^{n-k} \quad \text{for } 0 \leqslant k \leqslant n$$

Moments: $E[S_n] = n \cdot p$ and $Var[S_n] = n \cdot p \cdot (1-p)$.

Geometric distribution

Let r.v. T_i be the number of steps between increments of counting process S_n. Then:

$$Pr\{ T_i = k \} = (1-p)^{k-1} \cdot p \quad \text{for } k \geqslant 1$$

This is a *geometric distribution*. We have $E[T_i] = \frac{1}{p}$ and $Var[T_i] = \frac{1-p}{p^2}$.

Intuition: Geometric distribution = number of Bernoulli trials needed for one success.

Geometric distribution

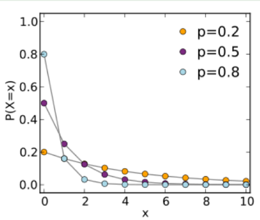
Geometric distribution

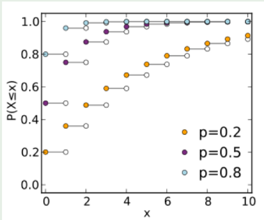
Let X be a discrete random variable, natural $k > 0$ and $0 < p \leqslant 1$. The mass function of a *geometric distribution* is given by:

$$Pr\{ X = k \} = (1-p)^{k-1} \cdot p$$

We have $E[X] = \frac{1}{p}$ and $Var[X] = \frac{1-p}{p^2}$ and cdf $Pr\{ X \leqslant k \} = 1 - (1-p)^k$.

Geometric distributions and their cdf's





Memoryless property

Theorem

1. For any random variable X with a geometric distribution:

$$Pr\{X = k + m \mid X > m\} \;=\; Pr\{X = k\} \quad \text{for any} \quad m \in T, k \geqslant 1$$

This is called the memoryless property, and X is a memoryless r.v..

2. Any discrete random variable which is memoryless is geometrically distributed.

Proof:

On the black board.

Joint distribution function

Joint distribution function

The *joint* distribution function of stochastic process $X = \{ X_t \mid t \in T \}$ is given for n, $t_1, \ldots, t_n \in T$ and $d_1, \ldots, d_n$ by:

$$F_X(d_1, \ldots, d_n; t_1, \ldots, t_n) = Pr\{ X(t_1) \leqslant d_1, \ldots, X(t_n) \leqslant d_n \}$$

The shape of F_X depends on the stochastic dependency between $X(t_i)$.

Stochastic independence

Random variables X_i on probability space $\mathcal{P}$ are *independent* if:

$$F_X(d_1, \ldots, d_n; t_1, \ldots, t_n) \;=\; \prod_{i=1}^{n} F_X(d_i; t_i) \;=\; \prod_{i=1}^{n} Pr\{ X(t_i) \leqslant d_i \}.$$

A renewal process is a discrete-time stochastic process where $X(t_1), X(t_2), \ldots$ are independent, identically distributed, non-negative random variables.