Part I

Basics of Probability Theory

Chapter 3. PROBABILITY THEORY BASICS

CHAPTER 3:
BASICS of PROBABILITY THEORY

PROBABILITY INTUITIVELY

PROBABILITY INTUITIVELY

Intuitively, **probability of an event** E is the ratio between the number of favorable elementary events involved in E to the number of all possible elementary events involved in E.

PROBABILITY INTUITIVELY

Intuitively, **probability of an event** E is the ratio between the number of favorable elementary events involved in E to the number of all possible elementary events involved in E.

$$Pr(E) = \frac{\text{number of favorable elementary events involved in } E}{\text{number of all possible elementary events involved in } E}$$

Example: Probability that when tossing a perfect 6-sided dice we get a number divided by 3 is

PROBABILITY INTUITIVELY

Intuitively, **probability of an event E** is the ratio between the number of favorable elementary events involved in E to the number of all possible elementary events involved in E.

$$Pr(E) = \frac{\text{number of favorable elementary events involved in } E}{\text{number of all possible elementary events involved in } E}$$

Example: Probability that when tossing a perfect 6-sided dice we get a number divided by 3 is

$$2/6 = 1/3$$

.

PROBABILITY INTUITIVELY

Intuitively, **probability of an event** E is the ratio between the number of favorable elementary events involved in E to the number of all possible elementary events involved in E.

$$Pr(E) = \frac{\text{number of favorable elementary events involved in } E}{\text{number of all possible elementary events involved in } E}$$

Example: Probability that when tossing a perfect 6-sided dice we get a number divided by 3 is

$$2/6 = 1/3$$

.

Key fact: Any probabilistic statement must refer to a specific underlying probability space - a space of elements to which a probability is assigned.

PROBABILITY SPACES

PROBABILITY SPACES

A **probability space** is defined in terms of a **sample space** Ω (often with an algebraic structure – for example outcomes of some cube tossing) and a **probability measure** (*probability distribution*) defined on Ω.

PROBABILITY SPACES

A **probability space** is defined in terms of a **sample space** Ω (often with an algebraic structure – for example outcomes of some cube tossing) and a **probability measure** (*probability distribution*) defined on Ω.

Subsets of a sample space Ω are called **events**. Elements of Ω are referred to as **elementary events**.

PROBABILITY SPACES

A **probability space** is defined in terms of a **sample space** Ω (often with an algebraic structure – for example outcomes of some cube tossing) and a **probability measure** (*probability distribution*) defined on Ω.

Subsets of a sample space Ω are called **events**. Elements of Ω are referred to as **elementary events**.

Intuitively, the sample space represents the set of all possible outcomes of a **probabilistic experiment** – for example of a cube tossing. An event represents a collection (a subset) of possible outcomes.

PROBABILITY SPACES

A **probability space** is defined in terms of a **sample space** Ω (often with an algebraic structure – for example outcomes of some cube tossing) and a **probability measure** (*probability distribution*) defined on Ω.

Subsets of a sample space Ω are called **events**. Elements of Ω are referred to as **elementary events**.

Intuitively, the sample space represents the set of all possible outcomes of a **probabilistic experiment** – for example of a cube tossing. An event represents a collection (a subset) of possible outcomes.

Intuitively - again, probability of an event E is the ratio between the number of favorable elementary events involved in E and the number of all possible elementary events.

PROBABILITY THEORY

PROBABILITY THEORY

Probability theory took almost 300 years to develop

PROBABILITY THEORY

Probability theory took almost 300 years to develop

from intuitive ideas of Pascal, Fermat and Huygens, around 1650,

PROBABILITY THEORY

Probability theory took almost 300 years to develop

from intuitive ideas of Pascal, Fermat and Huygens, around 1650,

to the currently acceptable axiomatic definition of probability (due to A. N. Kolmogorov in 1933).

AXIOMATIC APPROACH - I.

AXIOMATIC APPROACH - I.

Axiomatic approach: Probability distribution on a set Ω is every function $\Pr : 2^{\Omega} \rightarrow [0,1]$, satisfying the following axioms (of Kolmogorov):

AXIOMATIC APPROACH - I.

Axiomatic approach: Probability distribution on a set Ω is every function $\Pr : 2^{\Omega} \rightarrow [0, 1]$, satisfying the following axioms (of Kolmogorov):

1. $\Pr(\{x\}) \geq 0$ for any element (elementary event) $x \in \Omega$;

AXIOMATIC APPROACH - I.

Axiomatic approach: Probability distribution on a set Ω is every function $\Pr : 2^{\Omega} \rightarrow [0, 1]$, satisfying the following axioms (of Kolmogorov):

1. $\Pr(\{x\}) \geq 0$ for any element (elementary event) $x \in \Omega$;
2. $\Pr(\Omega) = 1$

AXIOMATIC APPROACH - I.

Axiomatic approach: Probability distribution on a set Ω is every function $\Pr : 2^{\Omega} \rightarrow [0,1]$, satisfying the following axioms (of Kolmogorov):

1. $\Pr(\{x\}) \geq 0$ for any element (elementary event) $x \in \Omega$;
2. $\Pr(\Omega) = 1$
3. $\Pr(A \cup B) = \Pr(A) + \Pr(B)$ if $A, B \subseteq \Omega$, $A \cap B = \emptyset$.

AXIOMATIC APPROACH - I.

Axiomatic approach: Probability distribution on a set Ω is every function $\Pr : 2^\Omega \rightarrow [0, 1]$, satisfying the following axioms (of Kolmogorov):

1. $\Pr(\{x\}) \geq 0$ for any element (elementary event) $x \in \Omega$;
2. $\Pr(\Omega) = 1$
3. $\Pr(A \cup B) = \Pr(A) + \Pr(B)$ if $A, B \subseteq \Omega$, $A \cap B = \emptyset$.

Example: **Probabilistic experiment** – cube tossing;

AXIOMATIC APPROACH - I.

Axiomatic approach: Probability distribution on a set Ω is every function $\Pr : 2^{\Omega} \rightarrow [0, 1]$, satisfying the following axioms (of Kolmogorov):

1. $\Pr(\{x\}) \geq 0$ for any element (elementary event) $x \in \Omega$;
2. $\Pr(\Omega) = 1$
3. $\Pr(A \cup B) = \Pr(A) + \Pr(B)$ if $A, B \subseteq \Omega$, $A \cap B = \emptyset$.

Example: **Probabilistic experiment** – cube tossing; **elementary events** – outcomes of cube tossing;

AXIOMATIC APPROACH - I.

Axiomatic approach: Probability distribution on a set Ω is every function $\Pr : 2^{\Omega} \rightarrow [0,1]$, satisfying the following axioms (of Kolmogorov):

1. $\Pr(\{x\}) \geq 0$ for any element (elementary event) $x \in \Omega$;
2. $\Pr(\Omega) = 1$
3. $\Pr(A \cup B) = \Pr(A) + \Pr(B)$ if $A, B \subseteq \Omega$, $A \cap B = \emptyset$.

Example: **Probabilistic experiment** – cube tossing; **elementary events** – outcomes of cube tossing; **probability distribution** – $\{p_1, p_2, p_3, p_4, p_5, p_6\}$, $\sum_{i=1}^{6} p_i = 1$, where p_i is probability that i is the outcome of a particular cube tossing.

AXIOMATIC APPROACH - I.

Axiomatic approach: Probability distribution on a set Ω is every function $\Pr : 2^{\Omega} \rightarrow [0,1]$, satisfying the following axioms (of Kolmogorov):

1. $\Pr(\{x\}) \geq 0$ for any element (elementary event) $x \in \Omega$;
2. $\Pr(\Omega) = 1$
3. $\Pr(A \cup B) = \Pr(A) + \Pr(B)$ if $A, B \subseteq \Omega$, $A \cap B = \emptyset$.

Example: **Probabilistic experiment** – cube tossing; **elementary events** – outcomes of cube tossing; **probability distribution** – $\{p_1, p_2, p_3, p_4, p_5, p_6\}$, $\sum_{i=1}^{6} p_i = 1$, where p_i is probability that i is the outcome of a particular cube tossing.

In general, a sample space is an arbitrary set. However, often we need (wish) to consider only some (family) of all possible events of 2^{Ω}.

AXIOMATIC APPROACH - I.

Axiomatic approach: Probability distribution on a set Ω is every function $\Pr : 2^{\Omega} \rightarrow [0, 1]$, satisfying the following axioms (of Kolmogorov):

1. $\Pr(\{x\}) \geq 0$ for any element (elementary event) $x \in \Omega$;
2. $\Pr(\Omega) = 1$
3. $\Pr(A \cup B) = \Pr(A) + \Pr(B)$ if $A, B \subseteq \Omega$, $A \cap B = \emptyset$.

Example: **Probabilistic experiment** – cube tossing; **elementary events** – outcomes of cube tossing; **probability distribution** – $\{p_1, p_2, p_3, p_4, p_5, p_6\}$, $\sum_{i=1}^{6} p_i = 1$, where p_i is probability that i is the outcome of a particular cube tossing.

In general, a sample space is an arbitrary set. However, often we need (wish) to consider only some (family) of all possible events of 2^{Ω}.

The fact that not all collections of events lead to well-defined probability spaces leads to the concepts presented on the next slide.

AXIOMATIC APPROACH - II.

AXIOMATIC APPROACH - II.

Definition: A **σ-field** $(\Omega, \mathbf{F})$ consists of a sample space Ω and a collection $\mathbf{F}$ of subsets of Ω satisfying the following conditions:

AXIOMATIC APPROACH - II.

Definition: A **σ-field** $(\Omega, \mathbf{F})$ consists of a sample space Ω and a collection **F** of subsets of Ω satisfying the following conditions:

1. $\emptyset \in \mathbf{F}$

AXIOMATIC APPROACH - II.

Definition: A **σ-field** $(\Omega, \mathbf{F})$ consists of a sample space Ω and a collection $\mathbf{F}$ of subsets of Ω satisfying the following conditions:

1. $\emptyset \in \mathbf{F}$
2. $\varepsilon \in \mathbf{F} \Rightarrow \overline{\varepsilon} \in \mathbf{F}$

AXIOMATIC APPROACH - II.

Definition: A **σ-field** $(\Omega, \mathbf{F})$ consists of a sample space Ω and a collection **F** of subsets of Ω satisfying the following conditions:

1. $\emptyset \in \mathbf{F}$
2. $\varepsilon \in \mathbf{F} \Rightarrow \overline{\varepsilon} \in \mathbf{F}$
3. $\varepsilon_1, \varepsilon_2, \ldots \in \mathbf{F} \Rightarrow (\varepsilon_1 \cup \varepsilon_2 \cup \ldots) \in \mathbf{F}$

AXIOMATIC APPROACH - II.

Definition: A **σ-field** $(\Omega, \mathbf{F})$ consists of a sample space Ω and a collection $\mathbf{F}$ of subsets of Ω satisfying the following conditions:

1. $\emptyset \in \mathbf{F}$
2. $\varepsilon \in \mathbf{F} \Rightarrow \overline{\varepsilon} \in \mathbf{F}$
3. $\varepsilon_1, \varepsilon_2, \ldots \in \mathbf{F} \Rightarrow (\varepsilon_1 \cup \varepsilon_2 \cup \ldots) \in \mathbf{F}$

Consequence

A σ-field is closed under countable unions and intersections.

AXIOMATIC APPROACH - II.

Definition: A **σ-field** $(\Omega, \mathbf{F})$ consists of a sample space Ω and a collection $\mathbf{F}$ of subsets of Ω satisfying the following conditions:

1. $\emptyset \in \mathbf{F}$
2. $\varepsilon \in \mathbf{F} \Rightarrow \overline{\varepsilon} \in \mathbf{F}$
3. $\varepsilon_1, \varepsilon_2, \ldots \in \mathbf{F} \Rightarrow (\varepsilon_1 \cup \varepsilon_2 \cup \ldots) \in \mathbf{F}$

Consequence

A σ-field is closed under countable unions and intersections.

Definition: A **probability measure** (*distribution*) $Pr : \mathbf{F} \to \mathbf{R}^{\geq 0}$ on a σ-field $(\Omega,\mathbf{F})$ is a function satisfying conditions:

AXIOMATIC APPROACH - II.

Definition: A **σ-field** $(\Omega, \mathbf{F})$ consists of a sample space Ω and a collection $\mathbf{F}$ of subsets of Ω satisfying the following conditions:

1. $\emptyset \in \mathbf{F}$
2. $\varepsilon \in \mathbf{F} \Rightarrow \overline{\varepsilon} \in \mathbf{F}$
3. $\varepsilon_1, \varepsilon_2, \ldots \in \mathbf{F} \Rightarrow (\varepsilon_1 \cup \varepsilon_2 \cup \ldots) \in \mathbf{F}$

Consequence

A σ-field is closed under countable unions and intersections.

Definition: A **probability measure** (*distribution*) $Pr : \mathbf{F} \rightarrow \mathbf{R}^{\geq 0}$ on a σ-field $(\Omega, \mathbf{F})$ is a function satisfying conditions:

1. If $\varepsilon \in \mathbf{F}$, then $0 \leq Pr(\varepsilon) \leq 1$.

AXIOMATIC APPROACH - II.

Definition: A **σ-field** $(\Omega, \mathbf{F})$ consists of a sample space Ω and a collection $\mathbf{F}$ of subsets of Ω satisfying the following conditions:

1. $\emptyset \in \mathbf{F}$
2. $\varepsilon \in \mathbf{F} \Rightarrow \overline{\varepsilon} \in \mathbf{F}$
3. $\varepsilon_1, \varepsilon_2, \ldots \in \mathbf{F} \Rightarrow (\varepsilon_1 \cup \varepsilon_2 \cup \ldots) \in \mathbf{F}$

Consequence

A σ-field is closed under countable unions and intersections.

Definition: A **probability measure** (*distribution*) $Pr : \mathbf{F} \to \mathbf{R}^{\geq 0}$ on a σ-field $(\Omega,\mathbf{F})$ is a function satisfying conditions:

1. If $\varepsilon \in \mathbf{F}$, then $0 \leq Pr(\varepsilon) \leq 1$.
2. $Pr[\Omega] = 1$.

AXIOMATIC APPROACH - II.

Definition: A **σ-field** $(\Omega, \mathbf{F})$ consists of a sample space Ω and a collection $\mathbf{F}$ of subsets of Ω satisfying the following conditions:

1. $\emptyset \in \mathbf{F}$
2. $\varepsilon \in \mathbf{F} \Rightarrow \overline{\varepsilon} \in \mathbf{F}$
3. $\varepsilon_1, \varepsilon_2, \ldots \in \mathbf{F} \Rightarrow (\varepsilon_1 \cup \varepsilon_2 \cup \ldots) \in \mathbf{F}$

Consequence

A σ-field is closed under countable unions and intersections.

Definition: A **probability measure** (*distribution*) $Pr : \mathbf{F} \rightarrow \mathbf{R}^{\geq 0}$ on a σ-field $(\Omega,\mathbf{F})$ is a function satisfying conditions:

1. If $\varepsilon \in \mathbf{F}$, then $0 \leq Pr(\varepsilon) \leq 1$.
2. $Pr[\Omega] = 1$.
3. For mutually disjoint events $\varepsilon_1, \varepsilon_2, \ldots$
   $Pr\left[\bigcup_i \varepsilon_i\right] = \sum_i Pr(\varepsilon_i)$

AXIOMATIC APPROACH - II.

Definition: A **σ-field** $(\Omega, \mathbf{F})$ consists of a sample space Ω and a collection $\mathbf{F}$ of subsets of Ω satisfying the following conditions:

1. $\emptyset \in \mathbf{F}$
2. $\varepsilon \in \mathbf{F} \Rightarrow \overline{\varepsilon} \in \mathbf{F}$
3. $\varepsilon_1, \varepsilon_2, \ldots \in \mathbf{F} \Rightarrow (\varepsilon_1 \cup \varepsilon_2 \cup \ldots) \in \mathbf{F}$

Consequence

A σ-field is closed under countable unions and intersections.

Definition: A **probability measure** (*distribution*) $Pr : \mathbf{F} \to \mathbf{R}^{\geq 0}$ on a σ-field $(\Omega,\mathbf{F})$ is a function satisfying conditions:

1. If $\varepsilon \in \mathbf{F}$, then $0 \leq Pr(\varepsilon) \leq 1$.
2. $Pr[\Omega] = 1$.
3. For mutually disjoint events $\varepsilon_1, \varepsilon_2, \ldots$
   $Pr\left[\bigcup_i \varepsilon_i\right] = \sum_i Pr(\varepsilon_i)$

Definition: A **probability space** $(\Omega,\mathbf{F},Pr)$ consists of a σ-field $(\Omega,\mathbf{F})$ with a probability measure Pr defined on $(\Omega, \mathbf{F})$.

PROBABILITIES and their PROPERTIES - I.

PROBABILITIES and their PROPERTIES - I.

Properties (for arbitrary events ε_i):

$$Pr(\overline{\varepsilon}) \quad = \quad 1 - Pr(\varepsilon);$$

PROBABILITIES and their PROPERTIES - I.

Properties (for arbitrary events ε_i):

$$\begin{aligned} Pr(\overline{\varepsilon}) &= 1 - Pr(\varepsilon); \\ Pr(\varepsilon_1 \cup \varepsilon_2) &= Pr(\varepsilon_1) + Pr(\varepsilon_2) - Pr(\varepsilon_1 \cap \varepsilon_2); \end{aligned}$$

PROBABILITIES and their PROPERTIES - I.

Properties (for arbitrary events ε_i):

$$\begin{aligned} Pr(\overline{\varepsilon}) &= 1 - Pr(\varepsilon); \\ Pr(\varepsilon_1 \cup \varepsilon_2) &= Pr(\varepsilon_1) + Pr(\varepsilon_2) - Pr(\varepsilon_1 \cap \varepsilon_2); \\ Pr(\bigcup_{i\geq 1} \varepsilon_i) &\leq \sum_{i\geq 1} Pr(\varepsilon_i). \end{aligned}$$

PROBABILITIES and their PROPERTIES - I.

Properties (for arbitrary events ε_i):

$$\begin{aligned}
Pr(\overline{\varepsilon}) &= 1 - Pr(\varepsilon);\\
Pr(\varepsilon_1 \cup \varepsilon_2) &= Pr(\varepsilon_1) + Pr(\varepsilon_2) - Pr(\varepsilon_1 \cap \varepsilon_2);\\
Pr(\bigcup_{i\geq 1} \varepsilon_i) &\leq \sum_{i\geq 1} Pr(\varepsilon_i).
\end{aligned}$$

Definition: Conditional probability of an event ε_1 given an event ε_2 is defined by

$$Pr[\varepsilon_1|\varepsilon_2] = \frac{Pr[\varepsilon_1 \cap \varepsilon_2]}{Pr[\varepsilon_2]}$$

if $Pr[\varepsilon_2] > 0$.

PROBABILITIES and their PROPERTIES - I.

Properties (for arbitrary events ε_i):

$$\begin{aligned} Pr(\overline{\varepsilon}) &= 1 - Pr(\varepsilon); \\ Pr(\varepsilon_1 \cup \varepsilon_2) &= Pr(\varepsilon_1) + Pr(\varepsilon_2) - Pr(\varepsilon_1 \cap \varepsilon_2); \\ Pr(\bigcup_{i \geq 1} \varepsilon_i) &\leq \sum_{i \geq 1} Pr(\varepsilon_i). \end{aligned}$$

Definition: Conditional probability of an event ε_1 given an event ε_2 is defined by

$$Pr[\varepsilon_1 | \varepsilon_2] = \frac{Pr[\varepsilon_1 \cap \varepsilon_2]}{Pr[\varepsilon_2]}$$

if $Pr[\varepsilon_2] > 0$.

Theorem: Law of the total probability Let $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k$ be a **partition** of a sample space Ω. Then for any event ε

$$Pr[\varepsilon] = \sum_{i=1}^{k} Pr[\varepsilon | \varepsilon_i] \cdot Pr[\varepsilon_i]$$

EXAMPLE:

Let us consider tossing of two perfect dices with sides labelled by 1, 2, 3, 4, 5, 6.

EXAMPLE:

Let us consider tossing of two perfect dices with sides labelled by 1, 2, 3, 4, 5, 6. Let

ε_1 be the event that the reminder at the division of the sum of the outcomes of both dices when divided by 4 is 3, and

EXAMPLE:

Let us consider tossing of two perfect dices with sides labelled by 1, 2, 3, 4, 5, 6. Let

ε_1 be the event that the reminder at the division of the sum of the outcomes of both dices when divided by 4 is 3, and

ε_2 be the event that the outcome of the first cube is 4.

EXAMPLE:

Let us consider tossing of two perfect dices with sides labelled by 1, 2, 3, 4, 5, 6. Let

ε_1 be the event that the reminder at the division of the sum of the outcomes of both dices when divided by 4 is 3, and

ε_2 be the event that the outcome of the first cube is 4.
In such a case

$$Pr[\varepsilon_1|\varepsilon_2] =$$

EXAMPLE:

Let us consider tossing of two perfect dices with sides labelled by 1, 2, 3, 4, 5, 6. Let

ε_1 be the event that the reminder at the division of the sum of the outcomes of both dices when divided by 4 is 3, and

ε_2 be the event that the outcome of the first cube is 4.
In such a case

$$Pr[\varepsilon_1|\varepsilon_2] = \frac{Pr[\varepsilon_1 \cap \varepsilon_2]}{Pr[\varepsilon_2]} =$$

EXAMPLE:

Let us consider tossing of two perfect dices with sides labelled by 1, 2, 3, 4, 5, 6. Let

ε_1 be the event that the reminder at the division of the sum of the outcomes of both dices when divided by 4 is 3, and

ε_2 be the event that the outcome of the first cube is 4.
In such a case

$$Pr[\varepsilon_1|\varepsilon_2] = \frac{Pr[\varepsilon_1 \cap \varepsilon_2]}{Pr[\varepsilon_2]} = \frac{\frac{1}{36}}{\frac{1}{6}} =$$

EXAMPLE:

Let us consider tossing of two perfect dices with sides labelled by 1, 2, 3, 4, 5, 6. Let

ε_1 be the event that the reminder at the division of the sum of the outcomes of both dices when divided by 4 is 3, and

ε_2 be the event that the outcome of the first cube is 4.
In such a case

$$Pr[\varepsilon_1|\varepsilon_2] = \frac{Pr[\varepsilon_1 \cap \varepsilon_2]}{Pr[\varepsilon_2]} = \frac{\frac{1}{36}}{\frac{1}{6}} = \frac{1}{6}$$

PROBABILITIES and their PROPERTIES - II.

PROBABILITIES and their PROPERTIES - II.

Theorem: (**Bayes' Rule/Law**)

PROBABILITIES and their PROPERTIES - II.

Theorem: (Bayes' Rule/Law)

(a) $Pr(\varepsilon_1) \cdot Pr(\varepsilon_2|\varepsilon_1) = Pr(\varepsilon_2) \cdot Pr(\varepsilon_1|\varepsilon_2)$ basic equality

PROBABILITIES and their PROPERTIES - II.

Theorem: (Bayes' Rule/Law)

(a) $Pr(\varepsilon_1) \cdot Pr(\varepsilon_2|\varepsilon_1) = Pr(\varepsilon_2) \cdot Pr(\varepsilon_1|\varepsilon_2)$ basic equality

(b) $Pr(\varepsilon_2|\varepsilon_1) = \frac{Pr(\varepsilon_2)\, Pr(\varepsilon_1|\varepsilon_2)}{Pr(\varepsilon_1)}$ simple version

PROBABILITIES and their PROPERTIES - II.

Theorem: (Bayes' Rule/Law)

(a) $Pr(\varepsilon_1) \cdot Pr(\varepsilon_2|\varepsilon_1) = Pr(\varepsilon_2) \cdot Pr(\varepsilon_1|\varepsilon_2)$ basic equality

(b) $Pr(\varepsilon_2|\varepsilon_1) = \frac{Pr(\varepsilon_2)\, Pr(\varepsilon_1|\varepsilon_2)}{Pr(\varepsilon_1)}$ simple version

(c) $Pr[\varepsilon_0|\varepsilon] = \frac{Pr[\varepsilon_0 \cap \varepsilon]}{Pr[\varepsilon]} = \frac{Pr[\varepsilon|\varepsilon_0] \cdot Pr[\varepsilon_0]}{\sum_{i=1}^{k} Pr[\varepsilon|\varepsilon_i] \cdot Pr[\varepsilon_i]}.$ extended version

PROBABILITIES and their PROPERTIES - II.

Theorem: (Bayes' Rule/Law)

(a) $Pr(\varepsilon_1) \cdot Pr(\varepsilon_2|\varepsilon_1) = Pr(\varepsilon_2) \cdot Pr(\varepsilon_1|\varepsilon_2)$ basic equality

(b) $Pr(\varepsilon_2|\varepsilon_1) = \frac{Pr(\varepsilon_2)\, Pr(\varepsilon_1|\varepsilon_2)}{Pr(\varepsilon_1)}$ simple version

(c) $Pr[\varepsilon_0|\varepsilon] = \frac{Pr[\varepsilon_0 \cap \varepsilon]}{Pr[\varepsilon]} = \frac{Pr[\varepsilon|\varepsilon_0] \cdot Pr[\varepsilon_0]}{\sum_{i=1}^{k} Pr[\varepsilon|\varepsilon_i] \cdot Pr[\varepsilon_i]}.$ extended version

Definition: Independence

PROBABILITIES and their PROPERTIES - II.

Theorem: (Bayes' Rule/Law)

(a) $Pr(\varepsilon_1) \cdot Pr(\varepsilon_2|\varepsilon_1) = Pr(\varepsilon_2) \cdot Pr(\varepsilon_1|\varepsilon_2)$ basic equality

(b) $Pr(\varepsilon_2|\varepsilon_1) = \frac{Pr(\varepsilon_2)\,Pr(\varepsilon_1|\varepsilon_2)}{Pr(\varepsilon_1)}$ simple version

(c) $Pr[\varepsilon_0|\varepsilon] = \frac{Pr[\varepsilon_0 \cap \varepsilon]}{Pr[\varepsilon]} = \frac{Pr[\varepsilon|\varepsilon_0] \cdot Pr[\varepsilon_0]}{\sum_{i=1}^{k} Pr[\varepsilon|\varepsilon_i] \cdot Pr[\varepsilon_i]}$. extended version

Definition: Independence

1. Two events $\varepsilon_1, \varepsilon_2$ are called **independent** if

$$Pr(\varepsilon_1 \cap \varepsilon_2) = Pr(\varepsilon_1) \cdot Pr(\varepsilon_2)$$

.

PROBABILITIES and their PROPERTIES - II.

Theorem: (Bayes' Rule/Law)

(a) $Pr(\varepsilon_1) \cdot Pr(\varepsilon_2|\varepsilon_1) = Pr(\varepsilon_2) \cdot Pr(\varepsilon_1|\varepsilon_2)$ basic equality

(b) $Pr(\varepsilon_2|\varepsilon_1) = \frac{Pr(\varepsilon_2)\,Pr(\varepsilon_1|\varepsilon_2)}{Pr(\varepsilon_1)}$ simple version

(c) $Pr[\varepsilon_0|\varepsilon] = \frac{Pr[\varepsilon_0 \cap \varepsilon]}{Pr[\varepsilon]} = \frac{Pr[\varepsilon|\varepsilon_0] \cdot Pr[\varepsilon_0]}{\sum_{i=1}^{k} Pr[\varepsilon|\varepsilon_i] \cdot Pr[\varepsilon_i]}.$ extended version

Definition: Independence

1. Two events $\varepsilon_1, \varepsilon_2$ are called **independent** if

$$Pr(\varepsilon_1 \cap \varepsilon_2) = Pr(\varepsilon_1) \cdot Pr(\varepsilon_2)$$

.

2. A collection of events $\{\varepsilon_i | i \in I\}$ is **independent** if for all subsets $S \subseteq I$

$$Pr\left[\bigcap_{i \in S} \varepsilon_i\right] = \prod_{i \in S} Pr[\varepsilon_i].$$

MODERN (BAYESIAN) INTERPRETATION of BAYES RULE

for the entire process of learning from evidence has the form

$$Pr[\varepsilon_1|\varepsilon] = \frac{Pr[\varepsilon_1 \cap \varepsilon]}{Pr[\varepsilon]} = \frac{Pr[\varepsilon|\varepsilon_1] \cdot Pr[\varepsilon_1]}{\sum_{i=1}^{k} Pr[\varepsilon|\varepsilon_i] \cdot Pr[\varepsilon_i]}.$$

MODERN (BAYESIAN) INTERPRETATION of BAYES RULE

for the entire process of learning from evidence has the form

$$Pr[\varepsilon_1|\varepsilon] = \frac{Pr[\varepsilon_1 \cap \varepsilon]}{Pr[\varepsilon]} = \frac{Pr[\varepsilon|\varepsilon_1] \cdot Pr[\varepsilon_1]}{\sum_{i=1}^{k} Pr[\varepsilon|\varepsilon_i] \cdot Pr[\varepsilon_i]}.$$

In modern terms the last equation says that $Pr[\varepsilon_1|\varepsilon]$, the probability of a hypothesis ε_1 (given information ε), equals $\Pr(\varepsilon_1)$, our initial estimate of its probability, times $Pr[\varepsilon|\varepsilon_1]$, the probability of each new piece of information (under the hypothesis ε_1), divided by the sum of the probabilities of data in all possible hypothesis (ε_i).

TWO BASIC INTERPRETATIONS of PROBABILITY

TWO BASIC INTERPRETATIONS of PROBABILITY

In Frequentist interpretation, probability is defined with respect to a large number of trials, each producing one outcome from a set of possible outcomes - the probability of an event A , Pr(A), is a proportion of trials producing an outcome in A.

TWO BASIC INTERPRETATIONS of PROBABILITY

In Frequentist interpretation, probability is defined with respect to a large number of trials, each producing one outcome from a set of possible outcomes - the probability of an event A , Pr(A), is a proportion of trials producing an outcome in A.

In Bayesian interpretation, probability measures a degree of belief. Bayes' theorem then links the degree of belief in a proposition before and after receiving an additional evidence that the proposition holds.

EXAMPLE 1

EXAMPLE 1

Let us toss a two regular cubes, one after another and let

EXAMPLE 1

Let us toss a two regular cubes, one after another and let

ε_1 be the event that the sum of both tosses is ≥ 10

EXAMPLE 1

Let us toss a two regular cubes, one after another and let

ε_1 be the event that the sum of both tosses is ≥ 10

ε_2 be the event that the first toss provides 5

EXAMPLE 1

Let us toss a two regular cubes, one after another and let

$$\varepsilon_1 \text{ be the event that the sum of both tosses is } \geq 10$$

$$\varepsilon_2 \text{ be the event that the first toss provides 5}$$

How much are: $Pr(\varepsilon_1), Pr(\varepsilon_2), Pr(\varepsilon_1|\varepsilon_2), Pr(\varepsilon_1 \cap \varepsilon_2)$?

EXAMPLE 1

Let us toss a two regular cubes, one after another and let

ε_1 be the event that the sum of both tosses is ≥ 10

ε_2 be the event that the first toss provides 5

How much are: $Pr(\varepsilon_1), Pr(\varepsilon_2), Pr(\varepsilon_1|\varepsilon_2), Pr(\varepsilon_1 \cap \varepsilon_2)$?

$$Pr(\varepsilon_1) =$$

EXAMPLE 1

Let us toss a two regular cubes, one after another and let

ε_1 be the event that the sum of both tosses is ≥ 10

ε_2 be the event that the first toss provides 5

How much are: $Pr(\varepsilon_1), Pr(\varepsilon_2), Pr(\varepsilon_1|\varepsilon_2), Pr(\varepsilon_1 \cap \varepsilon_2)$?

$$Pr(\varepsilon_1) = \frac{6}{36}$$

EXAMPLE 1

Let us toss a two regular cubes, one after another and let

ε_1 be the event that the sum of both tosses is ≥ 10

ε_2 be the event that the first toss provides 5

How much are: $Pr(\varepsilon_1), Pr(\varepsilon_2), Pr(\varepsilon_1|\varepsilon_2), Pr(\varepsilon_1 \cap \varepsilon_2)$?

$$Pr(\varepsilon_1) = \frac{6}{36}$$

$$Pr(\varepsilon_2) =$$

EXAMPLE 1

Let us toss a two regular cubes, one after another and let

ε_1 be the event that the sum of both tosses is ≥ 10

ε_2 be the event that the first toss provides 5

How much are: $Pr(\varepsilon_1), Pr(\varepsilon_2), Pr(\varepsilon_1|\varepsilon_2), Pr(\varepsilon_1 \cap \varepsilon_2)$?

$$Pr(\varepsilon_1) = \frac{6}{36}$$

$$Pr(\varepsilon_2) = \frac{1}{6}$$

EXAMPLE 1

Let us toss a two regular cubes, one after another and let

ε_1 be the event that the sum of both tosses is ≥ 10

ε_2 be the event that the first toss provides 5

How much are: $Pr(\varepsilon_1), Pr(\varepsilon_2), Pr(\varepsilon_1|\varepsilon_2), Pr(\varepsilon_1 \cap \varepsilon_2)$?

$$Pr(\varepsilon_1) = \frac{6}{36}$$

$$Pr(\varepsilon_2) = \frac{1}{6}$$

$$Pr(\varepsilon_1|\varepsilon_2) =$$

EXAMPLE 1

Let us toss a two regular cubes, one after another and let

ε_1 be the event that the sum of both tosses is ≥ 10

ε_2 be the event that the first toss provides 5

How much are: $Pr(\varepsilon_1), Pr(\varepsilon_2), Pr(\varepsilon_1|\varepsilon_2), Pr(\varepsilon_1 \cap \varepsilon_2)$?

$$Pr(\varepsilon_1) = \frac{6}{36}$$

$$Pr(\varepsilon_2) = \frac{1}{6}$$

$$Pr(\varepsilon_1|\varepsilon_2) = \frac{2}{6}$$

EXAMPLE 1

Let us toss a two regular cubes, one after another and let

ε_1 be the event that the sum of both tosses is ≥ 10

ε_2 be the event that the first toss provides 5

How much are: $Pr(\varepsilon_1), Pr(\varepsilon_2), Pr(\varepsilon_1|\varepsilon_2), Pr(\varepsilon_1 \cap \varepsilon_2)$?

$$Pr(\varepsilon_1) = \frac{6}{36}$$

$$Pr(\varepsilon_2) = \frac{1}{6}$$

$$Pr(\varepsilon_1|\varepsilon_2) = \frac{2}{6}$$

$$Pr(\varepsilon_1 \cap \varepsilon_2) =$$

EXAMPLE 1

Let us toss a two regular cubes, one after another and let

ε_1 be the event that the sum of both tosses is ≥ 10

ε_2 be the event that the first toss provides 5

How much are: $Pr(\varepsilon_1), Pr(\varepsilon_2), Pr(\varepsilon_1|\varepsilon_2), Pr(\varepsilon_1 \cap \varepsilon_2)$?

$$Pr(\varepsilon_1) = \frac{6}{36}$$

$$Pr(\varepsilon_2) = \frac{1}{6}$$

$$Pr(\varepsilon_1|\varepsilon_2) = \frac{2}{6}$$

$$Pr(\varepsilon_1 \cap \varepsilon_2) = \frac{2}{36}$$

EXAMPLE 2

EXAMPLE 2

Three coins are given - two fair ones and in the third one heads land with probability $2/3$, but we do not know which one is not fair one.

EXAMPLE 2

Three coins are given - two fair ones and in the third one heads land with probability $2/3$, but we do not know which one is not fair one.

When making an experiment and flipping all coins let the first two come up heads and the third one comes up tails.

EXAMPLE 2

Three coins are given - two fair ones and in the third one heads land with probability $2/3$, but we do not know which one is not fair one.

When making an experiment and flipping all coins let the first two come up heads and the third one comes up tails. What is probability that the first coin is the biased one?

EXAMPLE 2

Three coins are given - two fair ones and in the third one heads land with probability $2/3$, but we do not know which one is not fair one.

When making an experiment and flipping all coins let the first two come up heads and the third one comes up tails. What is probability that the first coin is the biased one?

Let ε_i be the event that the *i*th coin is biased and B be the event that three coins flips came up heads, heads, tails.

EXAMPLE 2

Three coins are given - two fair ones and in the third one heads land with probability $2/3$, but we do not know which one is not fair one.

When making an experiment and flipping all coins let the first two come up heads and the third one comes up tails. What is probability that the first coin is the biased one?

Let ε_i be the event that the *i*th coin is biased and B be the event that three coins flips came up heads, heads, tails.

Before flipping coins we have $Pr(\varepsilon_i) = \frac{1}{3}$ for all i. After flipping coins we have

EXAMPLE 2

Three coins are given - two fair ones and in the third one heads land with probability $2/3$, but we do not know which one is not fair one.

When making an experiment and flipping all coins let the first two come up heads and the third one comes up tails. What is probability that the first coin is the biased one?

Let ε_i be the event that the ith coin is biased and B be the event that three coins flips came up heads, heads, tails.

Before flipping coins we have $Pr(\varepsilon_i) = \frac{1}{3}$ for all i. After flipping coins we have

$$Pr(B|\varepsilon_1) = Pr(B|\varepsilon_2) = \frac{2}{3}\frac{1}{2}\frac{1}{2} = \frac{1}{6} \quad Pr(B|\varepsilon_3) = \frac{1}{2}\frac{1}{2}\frac{1}{3} = \frac{1}{12}$$

EXAMPLE 2

Three coins are given - two fair ones and in the third one heads land with probability $2/3$, but we do not know which one is not fair one.

When making an experiment and flipping all coins let the first two come up heads and the third one comes up tails. What is probability that the first coin is the biased one?

Let ε_i be the event that the ith coin is biased and B be the event that three coins flips came up heads, heads, tails.

Before flipping coins we have $Pr(\varepsilon_i) = \frac{1}{3}$ for all i. After flipping coins we have

$$Pr(B|\varepsilon_1) = Pr(B|\varepsilon_2) = \frac{2}{3}\frac{1}{2}\frac{1}{2} = \frac{1}{6} \qquad Pr(B|\varepsilon_3) = \frac{1}{2}\frac{1}{2}\frac{1}{3} = \frac{1}{12}$$

and using Bayes' law we have

EXAMPLE 2

Three coins are given - two fair ones and in the third one heads land with probability $2/3$, but we do not know which one is not fair one.

When making an experiment and flipping all coins let the first two come up heads and the third one comes up tails. What is probability that the first coin is the biased one?

Let ε_i be the event that the ith coin is biased and B be the event that three coins flips came up heads, heads, tails.

Before flipping coins we have $Pr(\varepsilon_i) = \frac{1}{3}$ for all i. After flipping coins we have

$$Pr(B|\varepsilon_1) = Pr(B|\varepsilon_2) = \frac{2}{3}\frac{1}{2}\frac{1}{2} = \frac{1}{6} \qquad Pr(B|\varepsilon_3) = \frac{1}{2}\frac{1}{2}\frac{1}{3} = \frac{1}{12}$$

and using Bayes' law we have

$$Pr(\varepsilon_1|B) = \frac{Pr(B|\varepsilon_1)Pr(\varepsilon_1)}{\sum_{i=1}^{3} Pr(B|\varepsilon_i)Pr\left(\varepsilon_i\right)} = \frac{\frac{1}{6}\cdot\frac{1}{3}}{\frac{1}{6}\cdot\frac{1}{3}+\frac{1}{6}\cdot\frac{1}{3}+\frac{1}{12}\cdot\frac{1}{3}} = \frac{2}{5}$$

EXAMPLE 2

Three coins are given - two fair ones and in the third one heads land with probability $2/3$, but we do not know which one is not fair one.

When making an experiment and flipping all coins let the first two come up heads and the third one comes up tails. What is probability that the first coin is the biased one?

Let ε_i be the event that the ith coin is biased and B be the event that three coins flips came up heads, heads, tails.

Before flipping coins we have $Pr(\varepsilon_i) = \frac{1}{3}$ for all i. After flipping coins we have

$$Pr(B|\varepsilon_1) = Pr(B|\varepsilon_2) = \frac{2}{3}\frac{1}{2}\frac{1}{2} = \frac{1}{6} \quad Pr(B|\varepsilon_3) = \frac{1}{2}\frac{1}{2}\frac{1}{3} = \frac{1}{12}$$

and using Bayes' law we have

$$Pr(\varepsilon_1|B) = \frac{Pr(B|\varepsilon_1)Pr(\varepsilon_1)}{\sum_{i=1}^{3} Pr(B|\varepsilon_i)Pr\left(\varepsilon_i\right)} = \frac{\frac{1}{6}\cdot\frac{1}{3}}{\frac{1}{6}\cdot\frac{1}{3}+\frac{1}{6}\cdot\frac{1}{3}+\frac{1}{12}\cdot\frac{1}{3}} = \frac{2}{5}$$

Therefore, the above outcome of the three coin flips increased the likelihood that the first coin is biased from $1/3$ to $2/5$

THEOREM

THEOREM

Let A and B be two events and let $\Pr(B) \neq 0$.

THEOREM

Let A and B be two events and let $\Pr(B) \neq 0$. Events A and B are independent if and only if

$$\Pr(A|B) = \Pr(A).$$

THEOREM

Let A and B be two events and let $\Pr(B) \neq 0$. Events A and B are independent if and only if

$$\Pr(A|B) = \Pr(A).$$

Proof

THEOREM

Let A and B be two events and let $\Pr(B) \neq 0$. Events A and B are independent if and only if

$$\Pr(A|B) = \Pr(A).$$

Proof

- Assume that A and B are independent and $\Pr(B) \neq 0$.

THEOREM

Let A and B be two events and let $\Pr(B) \neq 0$. Events A and B are independent if and only if

$$\Pr(A|B) = \Pr(A).$$

Proof

- Assume that A and B are independent and $\Pr(B) \neq 0$. By definition we have

  $$\Pr(A \cap B) = \Pr(A) \cdot \Pr(B)$$

  and therefore

  $$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)} = \frac{\Pr(A) \cdot \Pr(B)}{\Pr(B)} = \Pr(A).$$

THEOREM

Let A and B be two events and let $\Pr(B) \neq 0$. Events A and B are independent if and only if

$$\Pr(A|B) = \Pr(A).$$

Proof

- Assume that A and B are independent and $\Pr(B) \neq 0$. By definition we have

  $$\Pr(A \cap B) = \Pr(A) \cdot \Pr(B)$$

  and therefore

  $$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)} = \frac{\Pr(A) \cdot \Pr(B)}{\Pr(B)} = \Pr(A).$$

- Assume that $\Pr(A|B) = \Pr(A)$ and $\Pr(B) \neq 0$. Then

  $$\Pr(A) = \Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)}$$

  and multiplying by $\Pr(B)$ we get

  $$\Pr(A \cap B) = \Pr(A) \cdot \Pr(B)$$

  and so A and B are independent.

SUMMARY

SUMMARY

- The notion of conditional probability, of A given B, was introduced in order to get an instrument for analyzing an experiment A when one has partial information B about the outcome of the experiment A before experiment has finished.

SUMMARY

- The notion of conditional probability, of A given B, was introduced in order to get an instrument for analyzing an experiment A when one has partial information B about the outcome of the experiment A before experiment has finished.
- **We say that two events A and B are independent if the probability of A is equal to the probability of A given B,**

SUMMARY

- The notion of conditional probability, of A given B, was introduced in order to get an instrument for analyzing an experiment A when one has partial information B about the outcome of the experiment A before experiment has finished.
- **We say that two events A and B are independent if the probability of A is equal to the probability of A given B,**
- Other fundamental instruments for analysis of probabilistic experiments are **random variables** as functions from the sample space to $\mathbf{R}$, and **expectation** of random variables as the weighted averages of the values of random variables.

MONTY HALL PARADOX

MONTY HALL PARADOX

Let us assume that you see three doors D1, D2 and D3 and you know that behind one door is a car and behind other two are goats.

MONTY HALL PARADOX

Let us assume that you see three doors D1, D2 and D3 and you know that behind one door is a car and behind other two are goats.

Let us assume that you get a chance to choose one door

MONTY HALL PARADOX

Let us assume that you see three doors D1, D2 and D3 and you know that behind one door is a car and behind other two are goats.

Let us assume that you get a chance to choose one door and if you choose the door with car behind the car will be yours,

MONTY HALL PARADOX

Let us assume that you see three doors D1, D2 and D3 and you know that behind one door is a car and behind other two are goats.

Let us assume that you get a chance to choose one door and if you choose the door with car behind the car will be yours, and if you choose the door with a goat behind you will have to milk that goat for

Let us now assume that you have chosen the door D1.

Let us now assume that you have chosen the door D1.

and let afterwords a moderator comes who knows where car is and opens one of the doors D_2 or D_3, say D2, and you see that the goat is in.

Let us now assume that you have chosen the door D1.

and let afterwords a moderator comes who knows where car is and opens one of the doors D_2 or D_3, say D2, and you see that the goat is in.

Let us assume that at that point you get a chance to change your choice of the door.

Let us now assume that you have chosen the door D1.

and let afterwords a moderator comes who knows where car is and opens one of the doors D_2 or D_3, say D2, and you see that the goat is in.

Let us assume that at that point you get a chance to change your choice of the door.
Should you do that?

Let C_1 denote the event that the car is behind the door D1.

Let C_1 denote the event that the car is behind the door D1.
Let C_3 denote the event that the car is behind the door D3.

Let C_1 denote the event that the car is behind the door D1.
Let C_3 denote the event that the car is behind the door D3.
Let M_2 denote the event that moderator opens the door D2.

Let C_1 denote the event that the car is behind the door D1.
Let C_3 denote the event that the car is behind the door D3.
Let M_2 denote the event that moderator opens the door D2.

Let us assume that the moderator chosen a door at random if goats were behind both doors he could open. In such a case we have

Let C_1 denote the event that the car is behind the door D1.
Let C_3 denote the event that the car is behind the door D3.
Let M_2 denote the event that moderator opens the door D2.

Let us assume that the moderator chosen a door at random if goats were behind both doors he could open. In such a case we have

$$Pr[C_1] = \frac{1}{3} = Pr[C_3], \quad Pr[M_2|C1] = \frac{1}{2}, \quad Pr[M_2|C_3] = 1$$

Let C_1 denote the event that the car is behind the door D1.
Let C_3 denote the event that the car is behind the door D3.
Let M_2 denote the event that moderator opens the door D2.

Let us assume that the moderator chosen a door at random if goats were behind both doors he could open. In such a case we have

$$Pr[C_1] = \frac{1}{3} = Pr[C_3], \quad Pr[M_2|C1] = \frac{1}{2}, \quad Pr[M_2|C_3] = 1$$

Then it holds

Let C_1 denote the event that the car is behind the door D1.
Let C_3 denote the event that the car is behind the door D3.
Let M_2 denote the event that moderator opens the door D2.

Let us assume that the moderator chosen a door at random if goats were behind both doors he could open. In such a case we have

$$Pr[C_1] = \frac{1}{3} = Pr[C_3], \quad Pr[M_2|C1] = \frac{1}{2}, \quad Pr[M_2|C_3] = 1$$

Then it holds

$$Pr[C_1|M_2] = \frac{Pr[M_2|C_1]Pr[C_1]}{Pr[M_2]} = \frac{Pr[M_2|C_1]Pr[C_1]}{Pr[M_2|C_1]Pr[C_1] + Pr[M_2|C_3]Pr[C_3]} = \frac{1/6}{1/6 + 1/3}$$

Let C_1 denote the event that the car is behind the door D1.
Let C_3 denote the event that the car is behind the door D3.
Let M_2 denote the event that moderator opens the door D2.

Let us assume that the moderator chosen a door at random if goats were behind both doors he could open. In such a case we have

$$Pr[C_1] = \frac{1}{3} = Pr[C_3], \quad Pr[M_2|C1] = \frac{1}{2}, \quad Pr[M_2|C_3] = 1$$

Then it holds

$$Pr[C_1|M_2] = \frac{Pr[M_2|C_1]Pr[C_1]}{Pr[M_2]} = \frac{Pr[M_2|C_1]Pr[C_1]}{Pr[M_2|C_1]Pr[C_1] + Pr[M_2|C_3]Pr[C_3]} = \frac{1/6}{1/6+1/3}$$

Similarly

$$Pr[C_3|M_2] = \frac{Pr[M_2|C_3]Pr[C_3]}{Pr[M_2]} = \frac{Pr[M_2|C_3]Pr[C_3]}{Pr[M_2|C_1]Pr[C_1] + Pr[M_2|C_3]Pr[C_3]} = \frac{1/3}{1/6+1/3}$$

RANDOM VARIABLES - INFORMAL APPROACH

RANDOM VARIABLES - INFORMAL APPROACH

A random variable is a function defined on the elementary events of a probability space and having as values real numbers.

RANDOM VARIABLES - INFORMAL APPROACH

A random variable is a function defined on the elementary events of a probability space and having as values real numbers.

Example: In case of two tosses of a fair six-sided dice, the value of a random variable V can be the sum of the numbers on te two top spots on the dice rolls.

RANDOM VARIABLES - INFORMAL APPROACH

A random variable is a function defined on the elementary events of a probability space and having as values real numbers.

Example: In case of two tosses of a fair six-sided dice, the value of a random variable V can be the sum of the numbers on te two top spots on the dice rolls.

The value of V can therefore be an integer from the interval $[2, 12]$.

RANDOM VARIABLES - INFORMAL APPROACH

A random variable is a function defined on the elementary events of a probability space and having as values real numbers.

Example: In case of two tosses of a fair six-sided dice, the value of a random variable V can be the sum of the numbers on te two top spots on the dice rolls.

The value of V can therefore be an integer from the interval $[2, 12]$.

A random variable V with n potential values $v_1, v_2, \ldots, v_n$ is characterized by a probability distribution $p = (p_1, p_2, \ldots, p_n)$, where p_i is probability that V takes the value v_i.

RANDOM VARIABLES - INFORMAL APPROACH

A random variable is a function defined on the elementary events of a probability space and having as values real numbers.

Example: In case of two tosses of a fair six-sided dice, the value of a random variable V can be the sum of the numbers on te two top spots on the dice rolls.

The value of V can therefore be an integer from the interval $[2, 12]$.

A random variable V with n potential values $v_1, v_2, \ldots, v_n$ is characterized by a probability distribution $p = (p_1, p_2, \ldots, p_n)$, where p_i is probability that V takes the value v_i.

The concept of random variable is one of the most important of modern science and technology.

INDEPENDENCE of RANDOM VARIABLES

Definition Two random variables X, Y are called **independent random variables** if

$$x, y \in \mathbf{R} \Rightarrow Pr_{X,Y}(x, y) = Pr[X = x] \cdot Pr[Y = y]$$

EXPECTATION – MEAN of RANDOM VARIABLES

EXPECTATION – MEAN of RANDOM VARIABLES

Definition: The **expectation (mean** or **expected value) $\mathbf{E}[X]$** of a random variable X is defined as

$$\mathbf{E}[X] = \sum_{\omega \in \Omega} X(\omega) Pr_X(\omega).$$

Properties of he mean for random variabkes X and Y and a constant c:

EXPECTATION – MEAN of RANDOM VARIABLES

Definition: The **expectation (mean or expected value) $\mathbf{E}[X]$** of a random variable X is defined as

$$\mathbf{E}[X] = \sum_{\omega \in \Omega} X(\omega) Pr_X(\omega).$$

Properties of he mean for random variabkes X and Y and a constant c:

$$\mathbf{E}[X+Y] \quad = \quad \mathbf{E}[X] + \mathbf{E}[Y].$$

EXPECTATION – MEAN of RANDOM VARIABLES

Definition: The **expectation (mean or expected value) $\mathbf{E}[X]$** of a random variable X is defined as

$$\mathbf{E}[X] = \sum_{\omega \in \Omega} X(\omega) Pr_X(\omega).$$

Properties of he mean for random variabkes X and Y and a constant c:

$$\begin{aligned} \mathbf{E}[X+Y] &= \mathbf{E}[X] + \mathbf{E}[Y]. \\ \mathbf{E}[c \cdot X] &= c \cdot \mathbf{E}[X]. \end{aligned}$$

EXPECTATION – MEAN of RANDOM VARIABLES

Definition: The **expectation (mean or expected value)** $\mathbf{E}[X]$ of a random variable X is defined as

$$\mathbf{E}[X] = \sum_{\omega \in \Omega} X(\omega) Pr_X(\omega).$$

Properties of he mean for random variabkes X and Y and a constant c:

$$\begin{aligned}\mathbf{E}[X+Y] &= \mathbf{E}[X]+\mathbf{E}[Y].\\ \mathbf{E}[c\cdot X] &= c\cdot\mathbf{E}[X].\\ \mathbf{E}[X\cdot Y] &= \mathbf{E}[X]\cdot\mathbf{E}[Y], \qquad \text{if } X, Y \text{ are independent}\end{aligned}$$

The first of the above equalities is known as **linearity of expectations**. It can be extended to a finite number of random variables $X_1, \ldots, X_n$ to hold

$$\mathbf{E}[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} \mathbf{E}[X_i]$$

EXPECTATION – MEAN of RANDOM VARIABLES

Definition: The **expectation (mean or expected value) $\mathbf{E}[X]$** of a random variable X is defined as

$$\mathbf{E}[X] = \sum_{\omega \in \Omega} X(\omega) Pr_X(\omega).$$

Properties of he mean for random variabkes X and Y and a constant c:

$$\begin{aligned} \mathbf{E}[X+Y] &= \mathbf{E}[X] + \mathbf{E}[Y]. \\ \mathbf{E}[c \cdot X] &= c \cdot \mathbf{E}[X]. \\ \mathbf{E}[X \cdot Y] &= \mathbf{E}[X] \cdot \mathbf{E}[Y], \qquad \text{if } X, Y \text{ are independent} \end{aligned}$$

The first of the above equalities is known as **linearity of expectations**. It can be extended to a finite number of random variables $X_1, \ldots, X_n$ to hold

$$\mathbf{E}[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} \mathbf{E}[X_i]$$

and also to any countable set of random variables $X_1, X_2, \ldots$ to hold:

EXPECTATION – MEAN of RANDOM VARIABLES

Definition: The **expectation (mean or expected value)** $\mathbf{E}[X]$ of a random variable X is defined as

$$\mathbf{E}[X] = \sum_{\omega \in \Omega} X(\omega) Pr_X(\omega).$$

Properties of he mean for random variabkes X and Y and a constant c:

$$\begin{aligned} \mathbf{E}[X+Y] &= \mathbf{E}[X] + \mathbf{E}[Y]. \\ \mathbf{E}[c \cdot X] &= c \cdot \mathbf{E}[X]. \\ \mathbf{E}[X \cdot Y] &= \mathbf{E}[X] \cdot \mathbf{E}[Y], \qquad \text{if } X, Y \text{ are independent} \end{aligned}$$

The first of the above equalities is known as **linearity of expectations**. It can be extended to a finite number of random variables $X_1, \ldots, X_n$ to hold

$$\mathbf{E}[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} \mathbf{E}[X_i]$$

and also to any countable set of random variables $X_1, X_2, \ldots$ to hold: If $\sum_{i=1}^{\infty} \mathbf{E}[|X_i|] < \infty$, then $\sum_{i=1}^{\infty} |X_i| < \infty$ and

$$\mathbf{E}[\sum_{i=1}^{\infty} X_i] = \sum_{i=1}^{\infty} \mathbf{E}[X_i].$$

EXPECTATION VALUES

EXPECTATION VALUES

For any random variable X let $\mathbf{R}_X$ be the set of values of X. Using $\mathbf{R}_X$ one can show that

$$E[X] = \sum_{x \in \mathbf{R}_X} x \cdot Pr(X = x).$$

EXPECTATION VALUES

For any random variable X let $\mathbf{R}_X$ be the set of values of X. Using $\mathbf{R}_X$ one can show that

$$E[X] = \sum_{x \in \mathbf{R}_X} x \cdot Pr(X = x).$$

Using that one can show that for any real a, b it holds

$$\mathbf{E}[aX + b] \quad = \quad \sum_{x \in \mathbf{R}_X} (ax + b)\mathrm{Pr}(X = x)$$

EXPECTATION VALUES

For any random variable X let $\mathbf{R}_X$ be the set of values of X. Using $\mathbf{R}_X$ one can show that

$$E[X] = \sum_{x \in \mathbf{R}_X} x \cdot Pr(X = x).$$

Using that one can show that for any real a, b it holds

$$\begin{aligned}\mathbf{E}[aX + b] &= \sum_{x \in \mathbf{R}_X} (ax + b)\mathrm{Pr}(X = x) \\ &= a \sum_{x \in \mathbf{R}_X} x \cdot \mathrm{Pr}(X = x) + b \sum_{x \in \mathbf{R}_X} \mathrm{Pr}(X = x)\end{aligned}$$

EXPECTATION VALUES

For any random variable X let $\mathbf{R}_X$ be the set of values of X. Using $\mathbf{R}_X$ one can show that

$$E[X] = \sum_{x \in \mathbf{R}_X} x \cdot Pr(X = x).$$

Using that one can show that for any real a, b it holds

$$\begin{aligned}
\mathbf{E}[aX + b] &= \sum_{x \in \mathbf{R}_X} (ax + b)\mathrm{Pr}(X = x) \\
&= a \sum_{x \in \mathbf{R}_X} x \cdot \mathrm{Pr}(X = x) + b \sum_{x \in \mathbf{R}_X} \mathrm{Pr}(X = x) \\
&= a \cdot \mathbf{E}[X] + b
\end{aligned}$$

EXPECTATION VALUES

For any random variable X let $\mathbf{R}_X$ be the set of values of X. Using $\mathbf{R}_X$ one can show that

$$E[X] = \sum_{x \in \mathbf{R}_X} x \cdot Pr(X = x).$$

Using that one can show that for any real a, b it holds

$$\begin{aligned}
\mathbf{E}[aX + b] &= \sum_{x \in \mathbf{R}_X} (ax + b)\mathrm{Pr}(X = x) \\
&= a \sum_{x \in \mathbf{R}_X} x \cdot \mathrm{Pr}(X = x) + b \sum_{x \in \mathbf{R}_X} \mathrm{Pr}(X = x) \\
&= a \cdot \mathbf{E}[X] + b
\end{aligned}$$

The above relation is called **weak linearity of expectation**.

INDICATOR VARIABLES

INDICATOR VARIABLES

A random variable X is said to be an indicator variable if X takes on only values 1 and 0.

INDICATOR VARIABLES

A random variable X is said to be an indicator variable if X takes on only values 1 and 0.

For any set $A \subset S$, one can define an indicator variable X_A that takes value 1 on A and 0 on $S - A$, if $(S, \Pr)$ is the underlying probability space.

INDICATOR VARIABLES

A random variable X is said to be an indicator variable if X takes on only values 1 and 0.

For any set $A \subset S$, one can define an indicator variable X_A that takes value 1 on A and 0 on $S - A$, if $(S, \Pr)$ is the underlying probability space.

It holds:

INDICATOR VARIABLES

A random variable X is said to be an indicator variable if X takes on only values 1 and 0.

For any set $A \subset S$, one can define an indicator variable X_A that takes value 1 on A and 0 on $S - A$, if $(S, \Pr)$ is the underlying probability space.

It holds:

$$\begin{aligned} \mathbf{E}_{\mathbf{Pr}}[X_A] &= \sum_{s \in S} X_A(s) \cdot \Pr(\{s\}) \\ &= \sum_{s \in A} X_A(s) \cdot \Pr(\{s\}) + \sum_{s \in S-A} X_A(s) \cdot \Pr(\{s\}) \end{aligned}$$

INDICATOR VARIABLES

A random variable X is said to be an indicator variable if X takes on only values 1 and 0.

For any set $A \subset S$, one can define an indicator variable X_A that takes value 1 on A and 0 on $S - A$, if $(S, \Pr)$ is the underlying probability space.

It holds:

$$\begin{aligned}\mathbf{E}_{\mathbf{Pr}}[X_A] &= \sum_{s \in S} X_A(s) \cdot \Pr(\{s\}) \\ &= \sum_{s \in A} X_A(s) \cdot \Pr(\{s\}) + \sum_{s \in S-A} X_A(s) \cdot \Pr(\{s\}) \\ &= \sum_{s \in A} 1 \cdot \Pr(\{s\}) + \sum_{s \in S-A} 0 \cdot \Pr(\{s\})\end{aligned}$$

INDICATOR VARIABLES

A random variable X is said to be an indicator variable if X takes on only values 1 and 0.

For any set $A \subset S$, one can define an indicator variable X_A that takes value 1 on A and 0 on $S - A$, if $(S, \Pr)$ is the underlying probability space.

It holds:

$$\begin{aligned}
\mathbf{E}_{\mathbf{Pr}}[X_A] &= \sum_{s \in S} X_A(s) \cdot \Pr(\{s\}) \\
&= \sum_{s \in A} X_A(s) \cdot \Pr(\{s\}) + \sum_{s \in S-A} X_A(s) \cdot \Pr(\{s\}) \\
&= \sum_{s \in A} 1 \cdot \Pr(\{s\}) + \sum_{s \in S-A} 0 \cdot \Pr(\{s\}) \\
&= \sum_{s \in A} \Pr(\{s\})
\end{aligned}$$

INDICATOR VARIABLES

A random variable X is said to be an indicator variable if X takes on only values 1 and 0.

For any set $A \subset S$, one can define an indicator variable X_A that takes value 1 on A and 0 on $S - A$, if $(S, \Pr)$ is the underlying probability space.

It holds:

$$\begin{aligned}
\mathbf{E}_{\mathbf{Pr}}[X_A] &= \sum_{s \in S} X_A(s) \cdot \Pr(\{s\}) \\
&= \sum_{s \in A} X_A(s) \cdot \Pr(\{s\}) + \sum_{s \in S-A} X_A(s) \cdot \Pr(\{s\}) \\
&= \sum_{s \in A} 1 \cdot \Pr(\{s\}) + \sum_{s \in S-A} 0 \cdot \Pr(\{s\}) \\
&= \sum_{s \in A} \Pr(\{s\}) \\
&= \Pr(A)
\end{aligned}$$

VARIANCE and STANDARD DEVIATION

VARIANCE and STANDARD DEVIATION

Definition For a random variable X **variance** VX and **standard deviation** σX are defined by

$$\mathbf{V}X = \mathbf{E}((X - \mathbf{E}X)^2)$$

VARIANCE and STANDARD DEVIATION

Definition For a random variable X **variance** VX and **standard deviation** σX are defined by

$$\mathbf{V}X = \mathbf{E}((X - \mathbf{E}X)^2)$$

$$\sigma X = \sqrt{\mathbf{V}X}$$

VARIANCE and STANDARD DEVIATION

Definition For a random variable X **variance** VX and **standard deviation** σX are defined by

$$\mathbf{V}X = \mathbf{E}((X - \mathbf{E}X)^2)$$

$$\sigma X = \sqrt{\mathbf{V}X}$$

Since

$$\mathbf{E}((X - \mathbf{E}X)^2) \quad = \quad \mathbf{E}(X^2 - 2X\mathbf{E}X + (\mathbf{E}X)^2) =$$

VARIANCE and STANDARD DEVIATION

Definition For a random variable X **variance** VX and **standard deviation** σX are defined by

$$\mathbf{V}X = \mathbf{E}((X - \mathbf{E}X)^2)$$

$$\sigma X = \sqrt{\mathbf{V}X}$$

Since

$$\begin{aligned}\mathbf{E}((X - \mathbf{E}X)^2) &= \mathbf{E}(X^2 - 2X\mathbf{E}X + (\mathbf{E}X)^2) = \\ &= \mathbf{E}(X^2) - 2(\mathbf{E}X)^2 + (\mathbf{E}X)^2 =\end{aligned}$$

VARIANCE and STANDARD DEVIATION

Definition For a random variable X **variance** VX and **standard deviation** σX are defined by

$$\mathbf{V}X = \mathbf{E}((X - \mathbf{E}X)^2)$$

$$\sigma X = \sqrt{\mathbf{V}X}$$

Since

$$\begin{aligned}\mathbf{E}((X - \mathbf{E}X)^2) &= \mathbf{E}(X^2 - 2X\mathbf{E}X + (\mathbf{E}X)^2) = \\ &= \mathbf{E}(X^2) - 2(\mathbf{E}X)^2 + (\mathbf{E}X)^2 = \\ &= \mathbf{E}(X^2) - (\mathbf{E}X)^2,\end{aligned}$$

it holds

$$\mathbf{V}X = \mathbf{E}(X^2) - (\mathbf{E}X)^2$$

VARIANCE and STANDARD DEVIATION

Definition For a random variable X **variance** VX and **standard deviation** σX are defined by

$$\mathbf{V}X = \mathbf{E}((X - \mathbf{E}X)^2)$$

$$\sigma X = \sqrt{\mathbf{V}X}$$

Since

$$\begin{aligned} \mathbf{E}((X - \mathbf{E}X)^2) &= \mathbf{E}(X^2 - 2X\mathbf{E}X + (\mathbf{E}X)^2) = \\ &= \mathbf{E}(X^2) - 2(\mathbf{E}X)^2 + (\mathbf{E}X)^2 = \\ &= \mathbf{E}(X^2) - (\mathbf{E}X)^2, \end{aligned}$$

it holds

$$\mathbf{V}X = \mathbf{E}(X^2) - (\mathbf{E}X)^2$$

Example: Let $\Omega = \{1, 2, \ldots, 10\}$, $Pr(i) = \frac{1}{10}$, $X(i) = i$;

VARIANCE and STANDARD DEVIATION

Definition For a random variable X **variance** $\mathbf{V}X$ and **standard deviation** σX are defined by

$$\mathbf{V}X = \mathbf{E}((X - \mathbf{E}X)^2)$$

$$\sigma X = \sqrt{\mathbf{V}X}$$

Since

$$\begin{aligned}\mathbf{E}((X - \mathbf{E}X)^2) &= \mathbf{E}(X^2 - 2X\mathbf{E}X + (\mathbf{E}X)^2) = \\ &= \mathbf{E}(X^2) - 2(\mathbf{E}X)^2 + (\mathbf{E}X)^2 = \\ &= \mathbf{E}(X^2) - (\mathbf{E}X)^2,\end{aligned}$$

it holds

$$\mathbf{V}X = \mathbf{E}(X^2) - (\mathbf{E}X)^2$$

Example: Let $\Omega = \{1, 2, \ldots, 10\}$, $Pr(i) = \frac{1}{10}$, $X(i) = i$; $Y(i) = i - 1$ if $i \leq 5$ and $Y(i) = i + 1$ otherwise.

VARIANCE and STANDARD DEVIATION

Definition For a random variable X **variance** VX and **standard deviation** σX are defined by

$$\mathbf{V}X = \mathbf{E}((X - \mathbf{E}X)^2)$$

$$\sigma X = \sqrt{\mathbf{V}X}$$

Since

$$\begin{aligned} \mathbf{E}((X - \mathbf{E}X)^2) &= \mathbf{E}(X^2 - 2X\mathbf{E}X + (\mathbf{E}X)^2) = \\ &= \mathbf{E}(X^2) - 2(\mathbf{E}X)^2 + (\mathbf{E}X)^2 = \\ &= \mathbf{E}(X^2) - (\mathbf{E}X)^2, \end{aligned}$$

it holds

$$\mathbf{V}X = \mathbf{E}(X^2) - (\mathbf{E}X)^2$$

Example: Let $\Omega = \{1, 2, \ldots, 10\}$, $Pr(i) = \frac{1}{10}$, $X(i) = i$; $Y(i) = i - 1$ if $i \leq 5$ and $Y(i) = i + 1$ otherwise.
$\mathbf{E}X = \mathbf{E}Y = 5.5$, $\mathbf{E}(X^2) = \frac{1}{10}\sum_{i=1}^{10} i^2 = 38.5$, $\mathbf{E}(Y^2) = 44.5$; $\mathbf{V}X = 8.25$, $\mathbf{V}Y = 14.25$

TWO RULES

TWO RULES

For independent random variables X and Y and a real number c it holds

TWO RULES

For independent random variables X and Y and a real number c it holds

- $\mathbf{V}(cX) = c^2\mathbf{V}(X)$; $\sigma(cX) = c\sigma(X)$

TWO RULES

For independent random variables X and Y and a real number c it holds

- $\mathbf{V}(cX) = c^2\mathbf{V}(X)$; $\sigma(cX) = c\sigma(X)$
- $\mathbf{V}(X + Y) = \mathbf{V}(X) + \mathbf{V}(Y)$. $\sigma(X + Y) = \sqrt{V(X) + V(Y)}$.

MOMENTS

MOMENTS

Definition

For $k \in \mathbf{N}$ the k-th moment m_X^k and the k-th central moment μ_X^k of a random variable X are defined as follows

$$m_X^k \quad = \quad \mathbf{E}X^k$$

MOMENTS

Definition

For $k \in \mathbf{N}$ the k-th moment m_X^k and the k-th central moment μ_X^k of a random variable X are defined as follows

$$\begin{aligned} m_X^k &= \mathbf{E}X^k \\ \mu_X^k &= \mathbf{E}((X - \mathbf{E}X)^k) \end{aligned}$$

MOMENTS

Definition

For $k \in \mathbf{N}$ the k-th moment m_X^k and the k-th central moment μ_X^k of a random variable X are defined as follows

$$\begin{aligned} m_X^k &= \mathbf{E}X^k \\ \mu_X^k &= \mathbf{E}((X - \mathbf{E}X)^k) \end{aligned}$$

The **mean** of a random variable X is sometimes denoted by $\mu_X = m_X^1$ and its variance by μ_X^2.

EXAMPLE I

EXAMPLE I

Each week there is a lottery that always sells 100 tickets. One of the tickets wins 100 millions, all other tickets win nothing.

EXAMPLE I

Each week there is a lottery that always sells 100 tickets. One of the tickets wins 100 millions, all other tickets win nothing.

What is better: to buy in one week two tickets (Strategy I) or two tickets in two different weeks (Strategy II)?

EXAMPLE I

Each week there is a lottery that always sells 100 tickets. One of the tickets wins 100 millions, all other tickets win nothing.

What is better: to buy in one week two tickets (Strategy I) or two tickets in two different weeks (Strategy II)?

Or none of these two strategies is better than the second one?

EXAMPLE II

EXAMPLE II

With Strategy I we win (in millions)

EXAMPLE II

With Strategy I we win (in millions)

$$0 \text{ with probability } 0.98$$

EXAMPLE II

With Strategy I we win (in millions)

0 with probability 0.98

100 with probability 0.02

EXAMPLE II

With Strategy I we win (in millions)

0 with probability 0.98

100 with probability 0.02

With Strategy II we win (in millions)

EXAMPLE II

With Strategy I we win (in millions)

$$0 \text{ with probability } 0.98$$

$$100 \text{ with probability } 0.02$$

With Strategy II we win (in millions)

$$0 \text{ with probability } 0.9801 = 0.99 \cdot 0.99$$

EXAMPLE II

With Strategy I we win (in millions)

$$0 \text{ with probability } \; 0.98$$

$$100 \text{ with probability } \; 0.02$$

With Strategy II we win (in millions)

$$0 \text{ with probability } \; 0.9801 = 0.99 \cdot 0.99$$

$$100 \text{ with probability } \; 0.0198 = 2 \cdot 0.01 \cdot 0.99$$

EXAMPLE II

With Strategy I we win (in millions)

$$0 \text{ with probability } 0.98$$

$$100 \text{ with probability } 0.02$$

With Strategy II we win (in millions)

$$0 \text{ with probability } 0.9801 = 0.99 \cdot 0.99$$

$$100 \text{ with probability } 0.0198 = 2 \cdot 0.01 \cdot 0.99$$

$$200 \text{ with probability } 0.0001 = 0.01 \cdot 0.01$$

EXAMPLE II

With Strategy I we win (in millions)

$$0 \text{ with probability } 0.98$$

$$100 \text{ with probability } 0.02$$

With Strategy II we win (in millions)

$$0 \text{ with probability } 0.9801 = 0.99 \cdot 0.99$$

$$100 \text{ with probability } 0.0198 = 2 \cdot 0.01 \cdot 0.99$$

$$200 \text{ with probability } 0.0001 = 0.01 \cdot 0.01$$

Variance at Strategy I is 196

EXAMPLE II

With Strategy I we win (in millions)

$$0 \text{ with probability } 0.98$$

$$100 \text{ with probability } 0.02$$

With Strategy II we win (in millions)

$$0 \text{ with probability } 0.9801 = 0.99 \cdot 0.99$$

$$100 \text{ with probability } 0.0198 = 2 \cdot 0.01 \cdot 0.99$$

$$200 \text{ with probability } 0.0001 = 0.01 \cdot 0.01$$

Variance at Strategy I is 196

Variance at Strategy II is 198

PROBABILITY GENERATING FUNCTION

PROBABILITY GENERATING FUNCTION

The **probability density function** of a random variable X whose values are natural numbers **can be represented** by the following **probability generating function** (PGF):

$$G_X(z) = \sum_{k \geq 0} Pr(X = k) \cdot z^k.$$

PROBABILITY GENERATING FUNCTION

The **probability density function** of a random variable X whose values are natural numbers **can be represented** by the following **probability generating function** (PGF):

$$G_X(z) = \sum_{k \geq 0} Pr(X = k) \cdot z^k.$$

Main properties

PROBABILITY GENERATING FUNCTION

The **probability density function** of a random variable X whose values are natural numbers **can be represented** by the following **probability generating function** (PGF):

$$G_X(z) = \sum_{k \geq 0} Pr(X = k) \cdot z^k.$$

Main properties

$$G_X(1) = 1$$

PROBABILITY GENERATING FUNCTION

The **probability density function** of a random variable X whose values are natural numbers **can be represented** by the following **probability generating function** (PGF):

$$G_X(z) = \sum_{k \geq 0} Pr(X = k) \cdot z^k.$$

Main properties

$$G_X(1) = 1$$

$$\mathbf{EX} = \sum_{k \geq 0} k \cdot Pr(X = k) = \sum_{k \geq 0} Pr(X = k) \cdot (k \cdot 1^{k-1}) = \mathbf{G'_X(1)}.$$

PROBABILITY GENERATING FUNCTION

The **probability density function** of a random variable X whose values are natural numbers **can be represented** by the following **probability generating function** (PGF):

$$G_X(z) = \sum_{k \geq 0} Pr(X = k) \cdot z^k.$$

Main properties

$$G_X(1) = 1$$

$$\mathbf{EX} = \sum_{k \geq 0} k \cdot Pr(X = k) = \sum_{k \geq 0} Pr(X = k) \cdot (k \cdot 1^{k-1}) = \mathbf{G'_X(1)}.$$

Since it holds

$$\begin{aligned} \mathbf{E(X^2)} &= \sum_{k \geq 0} k^2 \cdot Pr(X = k) \\ &= \sum_{k \geq 0} Pr(X = k) \cdot (k \cdot (k-1) \cdot 1^{k-2} + k \cdot 1^{k-1}) \\ &= \mathbf{G''_X(1)} + \mathbf{G'_X}(1) \end{aligned}$$

PROBABILITY GENERATING FUNCTION

The **probability density function** of a random variable X whose values are natural numbers **can be represented** by the following **probability generating function** (PGF):

$$G_X(z) = \sum_{k \geq 0} Pr(X = k) \cdot z^k.$$

Main properties

$$G_X(1) = 1$$

$$\mathbf{EX} = \sum_{k \geq 0} k \cdot Pr(X = k) = \sum_{k \geq 0} Pr(X = k) \cdot (k \cdot 1^{k-1}) = \mathbf{G'_X(1)}.$$

Since it holds

$$\begin{aligned} \mathbf{E(X^2)} &= \sum_{k \geq 0} k^2 \cdot Pr(X = k) \\ &= \sum_{k \geq 0} Pr(X = k) \cdot (k \cdot (k-1) \cdot 1^{k-2} + k \cdot 1^{k-1}) \\ &= \mathbf{G''_X(1)} + \mathbf{G'_X}(1) \end{aligned}$$

we have

AN INTERPRETATION

AN INTERPRETATION

- Sometimes one can think of the expectation $\mathbf{E}[Y]$ of a random variable Y as the "best guess" or the "best prediction" of the value of Y.

AN INTERPRETATION

- Sometimes one can think of the expectation $\mathbf{E}[Y]$ of a random variable Y as the "best guess" or the "best prediction" of the value of Y.
- It is the "best guess" in the sense that among all constants m the expectation $\mathbf{E}[(Y-m)^2]$ is minimal when $m = \mathbf{E}[\mathbf{Y}]$.

WHY ARE PGF USEFUL?

WHY ARE PGF USEFUL?

Main reason: For many important probability distributions their PGF are very simple and easy to work with.

WHY ARE PGF USEFUL?

Main reason: For many important probability distributions their PGF are very simple and easy to work with.

For example, for the **uniform distribution** on the set $\{0, 1, \ldots, n-1\}$ the PGF has form

$$U_n(z) = \frac{1}{n}(1 + z + \ldots + z^{n-1}) = \frac{1}{n} \cdot \frac{1 - z^n}{1 - z}.$$

Problem is with the case $z = 1$.

PROPERTIES of GENERATING FUNCTIONS

PROPERTIES of GENERATING FUNCTIONS

Property 1 If $X_1, \ldots, X_k$ are independent random variables with PGFs $G_1(z), \ldots, G_k(z)$, then the random variable $Y = \sum_{i=1}^{k} X_i$ has as its PGF the function

$$G(z) = \prod_{i=1}^{k} G_i(z).$$

PROPERTIES of GENERATING FUNCTIONS

Property 1 If $X_1, \ldots, X_k$ are independent random variables with PGFs $G_1(z), \ldots, G_k(z)$, then the random variable $Y = \sum_{i=1}^{k} X_i$ has as its PGF the function

$$G(z) = \prod_{i=1}^{k} G_i(z).$$

Property 2 Let $X_1, \ldots, X_k$ be a sequence of independent random variables with the same PGF $G_X(z)$. If Y is a random variable with PGF $G_Y(z)$ and Y is independent of all X_i, then the random variable $S = X_1 + \ldots + X_Y$ has as PGF the function

$$G_S(z) = G_Y(G_X(z)).$$

IMPORTANT DISTRIBUTIONS

Two important distributions are connected with experiments, called **Bernoulli trials**, that have two possible outcomes:

IMPORTANT DISTRIBUTIONS

Two important distributions are connected with experiments, called **Bernoulli trials**, that have two possible outcomes:

- **success** with probability p

IMPORTANT DISTRIBUTIONS

Two important distributions are connected with experiments, called **Bernoulli trials**, that have two possible outcomes:

- **success** with probability p
- **failure** with probability $q = 1 - p$

IMPORTANT DISTRIBUTIONS

Two important distributions are connected with experiments, called **Bernoulli trials**, that have two possible outcomes:

- **success** with probability p
- **failure** with probability $q = 1 - p$

Coin tossing is an example of a Bernoulli trial.

1. Let values of a random variable X be the number of trials needed to obtain a success. Then

$$Pr(X = k) =$$

IMPORTANT DISTRIBUTIONS

Two important distributions are connected with experiments, called **Bernoulli trials**, that have two possible outcomes:

- **success** with probability p
- **failure** with probability $q = 1 - p$

Coin tossing is an example of a Bernoulli trial.

1. Let values of a random variable X be the number of trials needed to obtain a success. Then

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

Such a probability distribution is called the **geometric distribution** and such a variable geometric random variable.

IMPORTANT DISTRIBUTIONS

Two important distributions are connected with experiments, called **Bernoulli trials**, that have two possible outcomes:

- **success** with probability p
- **failure** with probability $q = 1 - p$

Coin tossing is an example of a Bernoulli trial.

1. Let values of a random variable X be the number of trials needed to obtain a success. Then

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

Such a probability distribution is called the **geometric distribution** and such a variable geometric random variable. It holds

$$\mathbf{E}X = \frac{1}{p} \qquad \mathbf{V}X = \frac{q}{p^2} \qquad G(z) = \frac{pz}{1-qz}$$

2. Let values of a random variable Y be the number of successes in n trials. Then

$$Pr(Y = k) =$$

IMPORTANT DISTRIBUTIONS

Two important distributions are connected with experiments, called **Bernoulli trials**, that have two possible outcomes:

- **success** with probability p
- **failure** with probability $q = 1 - p$

Coin tossing is an example of a Bernoulli trial.

1. Let values of a random variable X be the number of trials needed to obtain a success. Then

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

Such a probability distribution is called the **geometric distribution** and such a variable geometric random variable. It holds

$$\mathbf{E}X = \frac{1}{p} \qquad \mathbf{V}X = \frac{q}{p^2} \qquad G(z) = \frac{pz}{1 - qz}$$

2. Let values of a random variable Y be the number of successes in n trials. Then

$$Pr(Y = k) = \left(\begin{array}{c} n \\ k \end{array} \right) p^k q^{n-k}$$

Such a probability distribution is called the **binomial distribution** and it holds

$$\mathbf{E}Y = np \qquad \mathbf{V}Y = npq \qquad G(z) = (q + pz)^n$$

and also

$$\mathbf{E}Y^2 = n(n-1)p^2 + np$$

BERNOULLI DISTRIBUTION

BERNOULLI DISTRIBUTION

Let X be a binary random variable (called usually Bernoulli or indicator random variable) that takes value 1 with probability p and 0 with probability $q = 1 - p$, then it holds

BERNOULLI DISTRIBUTION

Let X be a binary random variable (called usually Bernoulli or indicator random variable) that takes value 1 with probability p and 0 with probability $q = 1 - p$, then it holds

$$\mathbf{E}[X] = p \qquad \mathbf{V}X = pq \qquad G[z] = q + pz.$$

BINOMIAL DISTRIBUTION revisited

BINOMIAL DISTRIBUTION revisited

Let $X_1, \ldots, X_n$ be random variables having Bernoulli distribution with the common parameter p.

BINOMIAL DISTRIBUTION revisited

Let $X_1, \ldots, X_n$ be random variables having Bernoulli distribution with the common parameter p.
The random variable

$$X = X_1 + X_2 + \ldots + X_n$$

has so called binomial distribution denoted $B(n, p)$ with the density function denoted

$$B(k, n, p) = Pr(X = k) = \left(\begin{array}{c} n \\ k \end{array} \right) p^k q^{(n-k)}$$

POISSON DISTRIBUTION

Poisson distribution

POISSON DISTRIBUTION

Poisson distribution

Let $\lambda \in \mathbf{R}^{>0}$. The Poisson distribution with the parameter λ is the probability distribution with the density function

$$p(x) = \begin{cases} \lambda^x \frac{e^{-\lambda}}{x!}, & \text{for } x = 0, 1, 2, ... \\ 0, & \text{otherwise} \end{cases}$$

POISSON DISTRIBUTION

Poisson distribution

Let $\lambda \in \mathbf{R}^{>0}$. The Poisson distribution with the parameter λ is the probability distribution with the density function

$$p(x) = \begin{cases} \lambda^x \frac{e^{-\lambda}}{x!}, & \text{for } x = 0, 1, 2, \ldots \\ 0, & \text{otherwise} \end{cases}$$

For large n the Poisson distribution is a good approximation to the Binomial distribution $B(n, \frac{\lambda}{n})$

POISSON DISTRIBUTION

Poisson distribution

Let $\lambda \in \mathbf{R}^{>0}$. The Poisson distribution with the parameter λ is the probability distribution with the density function

$$p(x) = \begin{cases} \lambda^x \frac{e^{-\lambda}}{x!}, & \text{for } x = 0, 1, 2, ... \\ 0, & \text{otherwise} \end{cases}$$

For large n the Poisson distribution is a good approximation to the Binomial distribution $B(n, \frac{\lambda}{n})$

Property of a Poisson random variable X:

$$\mathbf{E}[X] = \lambda \qquad \mathbf{V}X = \lambda \qquad G[z] = e^{\lambda(z-1)}$$

EXPECTATION+VARIANCE OF SUMS OF RANDOM VARIABLES

EXPECTATION+VARIANCE OF SUMS OF RANDOM VARIABLES

Let

$$S_n = \sum_{i=1}^{n} X_i$$

where each X_i is a random variable which takes on value 1 (0) with probability p $(1 - p = q)$.

EXPECTATION+VARIANCE OF SUMS OF RANDOM VARIABLES

Let

$$S_n = \sum_{i=1}^{n} X_i$$

where each X_i is a random variable which takes on value 1 (0) with probability p $(1 - p = q)$.
It clearly holds

$$\mathbf{E}(X_i) \quad = \quad p$$

EXPECTATION+VARIANCE OF SUMS OF RANDOM VARIABLES

Let

$$S_n = \sum_{i=1}^{n} X_i$$

where each X_i is a random variable which takes on value 1 (0) with probability p $(1 - p = q)$.
It clearly holds

$$\begin{aligned} \mathbf{E}(X_i) &= p \\ \mathbf{E}(X_i^2) &= p \end{aligned}$$

EXPECTATION+VARIANCE OF SUMS OF RANDOM VARIABLES

Let

$$S_n = \sum_{i=1}^{n} X_i$$

where each X_i is a random variable which takes on value 1 (0) with probability p $(1 - p = q)$.
It clearly holds

$$\begin{aligned} \mathbf{E}(X_i) &= p \\ \mathbf{E}(X_i^2) &= p \\ \mathbf{E}(S_n) &= \mathbf{E}(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} \mathbf{E}(X_i) = np \end{aligned}$$

EXPECTATION+VARIANCE OF SUMS OF RANDOM VARIABLES

Let

$$S_n = \sum_{i=1}^{n} X_i$$

where each X_i is a random variable which takes on value 1 (0) with probability p $(1 - p = q)$.
It clearly holds

$$\begin{aligned}
\mathbf{E}(X_i) &= p \\
\mathbf{E}(X_i^2) &= p \\
\mathbf{E}(S_n) &= \mathbf{E}(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} \mathbf{E}(X_i) = np \\
\mathbf{E}(S_n^2) &= \mathbf{E}((\sum_{i=1}^{n} X_i)^2) = \mathbf{E}(\sum_{i=1}^{n} X_i^2 + \sum_{i \neq j} X_i X_j) = \\
&= \sum^{n} \mathbf{E}(X_i^2) + \sum \mathbf{E}(X_i X_j)
\end{aligned}$$

Hence

$$\begin{aligned}\mathbf{E}(S_n^2) &= \mathbf{E}((\sum_{i=1}^{n} X_i)^2) = \mathbf{E}(\sum_{i=1}^{n} X_i^2 + \sum_{i\neq j} X_i X_j) = \\ &= \sum_{i=1}^{n} \mathbf{E}(X_i^2) + \sum_{i\neq j} \mathbf{E}(X_i X_j)\end{aligned}$$

Hence

$$\begin{aligned}\mathbf{E}(S_n^2) &= \mathbf{E}((\sum_{i=1}^{n} X_i)^2) = \mathbf{E}(\sum_{i=1}^{n} X_i^2 + \sum_{i\neq j} X_i X_j) = \\ &= \sum_{i=1}^{n} \mathbf{E}(X_i^2) + \sum_{i\neq j} \mathbf{E}(X_i X_j)\end{aligned}$$

and therefore, if X_i, X_j are pairwise independent, as in this case, $\mathbf{E}(X_i X_j) =$ $= \mathbf{E}(X_i)\mathbf{E}(X_j)$

Hence

$$\begin{aligned}\mathbf{E}(S_n^2) &= \mathbf{E}((\sum_{i=1}^{n} X_i)^2) = \mathbf{E}(\sum_{i=1}^{n} X_i^2 + \sum_{i\neq j} X_i X_j) = \\ &= \sum_{i=1}^{n} \mathbf{E}(X_i^2) + \sum_{i\neq j} \mathbf{E}(X_i X_j)\end{aligned}$$

and therefore, if X_i, X_j are pairwise independent, as in this case, $\mathbf{E}(X_i X_j) =$ $= \mathbf{E}(X_i)\mathbf{E}(X_j)$ Hence

$$\begin{aligned}\mathbf{E}(S_n^2) &= np + 2\left(\begin{array}{c} n \\ 2 \end{array}\right)p^2 \\ &= np + n(n-1)p^2 \\ &= np(1-p) + n^2p^2 \\ &= n^2p^2 + npq\end{aligned}$$

Hence

$$\begin{aligned}\mathbf{E}(S_n^2) &= \mathbf{E}((\sum_{i=1}^{n} X_i)^2) = \mathbf{E}(\sum_{i=1}^{n} X_i^2 + \sum_{i\neq j} X_i X_j) = \\ &= \sum_{i=1}^{n} \mathbf{E}(X_i^2) + \sum_{i\neq j} \mathbf{E}(X_i X_j)\end{aligned}$$

and therefore, if X_i, X_j are pairwise independent, as in this case, $\mathbf{E}(X_i X_j) =$ $= \mathbf{E}(X_i)\mathbf{E}(X_j)$ Hence

$$\begin{aligned}\mathbf{E}(S_n^2) &= np + 2\left(\begin{array}{c} n \\ 2 \end{array}\right) p^2 \\ &= np + n(n-1)p^2 \\ &= np(1-p) + n^2p^2 \\ &= n^2p^2 + npq \\ VAR[S_n] &= \mathbf{E}(S_n^2) - (\mathbf{E}(S_n))^2 = n^2p^2 + npq - n^2p^2 = npq\end{aligned}$$

MOMENT INEQUALITIES

MOMENT INEQUALITIES

The following inequality, and several of its special cases, play very important role in the analysis of randomized computations:

Let X be a random variable that takes on values x with probability $p(x)$.

MOMENT INEQUALITIES

The following inequality, and several of its special cases, play very important role in the analysis of randomized computations:

Let X be a random variable that takes on values x with probability $p(x)$.

Theorem For any $\lambda > 0$ the so called k^{th} **moment inequality** holds:

$$Pr[|X| > \lambda] \leq \frac{\mathbf{E}(|X|^k)}{\lambda^k}$$

MOMENT INEQUALITIES

The following inequality, and several of its special cases, play very important role in the analysis of randomized computations:

Let X be a random variable that takes on values x with probability $p(x)$.

Theorem For any $\lambda > 0$ the so called k^{th} **moment inequality** holds:

$$Pr[|X| > \lambda] \leq \frac{\mathbf{E}(|X|^k)}{\lambda^k}$$

Proof of the above inequality;

MOMENT INEQUALITIES

The following inequality, and several of its special cases, play very important role in the analysis of randomized computations:

Let X be a random variable that takes on values x with probability $p(x)$.

Theorem For any $\lambda > 0$ the so called k^{th} **moment inequality** holds:

$$Pr[|X| > \lambda] \leq \frac{\mathbf{E}(|X|^k)}{\lambda^k}$$

Proof of the above inequality;

$$\begin{aligned} \mathbf{E}(|X|^k) &= \sum |x|^k p(x) \geq \sum_{|x|>\lambda} |x|^k p(x) \geq \\ &\geq \lambda^k \sum_{|x|>\lambda} p(x) = \lambda^k Pr[|X| > \lambda] \end{aligned}$$

Two important special cases - I.1

of the moment inequality;

$$Pr[|X| > \lambda] \leq \frac{\mathbf{E}(|X|^k)}{\lambda^k}$$

Two important special cases - I.1

of the moment inequality;

$$Pr[|X| > \lambda] \leq \frac{\mathbf{E}(|X|^k)}{\lambda^k}$$

Case 1 $k \to 1 \qquad \lambda \to \lambda\mathbf{E}(|X|)$

$$Pr[|X| \geq \lambda\mathbf{E}(|X|)] \leq \frac{1}{\lambda}$$ **Markov's inequality**

Two important special cases - I.1

of the moment inequality;

$$Pr[|X| > \lambda] \leq \frac{\mathbf{E}(|X|^k)}{\lambda^k}$$

Case 1 $k \to 1 \qquad \lambda \to \lambda\mathbf{E}(|X|)$

$$Pr[|X| \geq \lambda\mathbf{E}(|X|)] \leq \frac{1}{\lambda}$$ **Markov′s inequality**

Case 2 $k \to 2 \qquad X \to X - \mathbf{E}(X), \lambda \to \lambda\sqrt{V(X)}$

Two important special cases - I.1

of the moment inequality;

$$Pr[|X| > \lambda] \leq \frac{\mathbf{E}(|X|^k)}{\lambda^k}$$

Case 1 $\quad k \to 1 \qquad \lambda \to \lambda \mathbf{E}(|X|)$

$$Pr[|X| \geq \lambda \mathbf{E}(|X|)] \leq \frac{1}{\lambda}$$ **Markov′s inequality**

Case 2 $\quad k \to 2 \qquad X \to X - \mathbf{E}(X), \lambda \to \lambda\sqrt{V(X)}$

$$Pr\left[|X - \mathbf{E}(X)| \geq \lambda\sqrt{V(X)}\right] \leq \frac{\mathbf{E}((X - \mathbf{E}(X))^2)}{\lambda^2 V(X)} =$$

$$= \frac{V(X)}{\lambda^2 V(X)} = \frac{1}{\lambda^2}$$ **Chebyshev′s inequality**

Two important special cases - I.1

of the moment inequality;

$$Pr[|X| > \lambda] \leq \frac{\mathbf{E}(|X|^k)}{\lambda^k}$$

Case 1 $k \to 1 \qquad \lambda \to \lambda\mathbf{E}(|X|)$

$$Pr[|X| \geq \lambda\mathbf{E}(|X|)] \leq \frac{1}{\lambda}$$ **Markov′s inequality**

Case 2 $k \to 2 \qquad X \to X - \mathbf{E}(X), \lambda \to \lambda\sqrt{V(X)}$

$$Pr\left[|X - \mathbf{E}(X)| \geq \lambda\sqrt{V(X)}\right] \leq \frac{\mathbf{E}((X - \mathbf{E}(X))^2)}{\lambda^2 V(X)} =$$

$$= \frac{V(X)}{\lambda^2 V(X)} = \frac{1}{\lambda^2}$$ **Chebyshev′s inequality**

Another variant of Chebyshev's inequality:

$$\Pr[|X - \mathbf{E}(X)| \geq \lambda] \leq \frac{V(X)}{\lambda^2}$$

Two important special cases - I.1

of the moment inequality;

$$Pr[|X| > \lambda] \leq \frac{\mathbf{E}(|X|^k)}{\lambda^k}$$

Case 1 $k \to 1 \qquad \lambda \to \lambda \mathbf{E}(|X|)$

$$Pr[|X| \geq \lambda \mathbf{E}(|X|)] \leq \frac{1}{\lambda}$$ **Markov′s inequality**

Case 2 $k \to 2 \qquad X \to X - \mathbf{E}(X), \lambda \to \lambda\sqrt{V(X)}$

$$Pr\left[|X - \mathbf{E}(X)| \geq \lambda\sqrt{V(X)}\right] \leq \frac{\mathbf{E}((X-\mathbf{E}(X))^2)}{\lambda^2 V(X)} =$$

$$= \frac{V(X)}{\lambda^2 V(X)} = \frac{1}{\lambda^2}$$ **Chebyshev′s inequality**

Another variant of Chebyshev's inequality:

$$\Pr[|X - \mathbf{E}(X)| \geq \lambda] \leq \frac{V(X)}{\lambda^2}$$

and this is one of the main reasons why variance is used.

Two important special cases - I.2

Two important special cases - I.2

The following generalization of the moment inequality is also of importance:

Two important special cases - I.2

The following generalization of the moment inequality is also of importance:

Theorem

If $g(x)$ *is non-decreasing on* $[0, \infty)$, *then*

$$Pr[|X| > \lambda] \leq \frac{\mathbf{E}(g(X))}{g(\lambda)}$$

Two important special cases - I.2

The following generalization of the moment inequality is also of importance:

Theorem

If $g(x)$ is non-decreasing on $[0, \infty)$, then

$$Pr[|X| > \lambda] \leq \frac{\mathbf{E}(g(X))}{g(\lambda)}$$

As a special case, namely if $g(x) = e^{tx}$, we get:

$$Pr[|X| > \lambda] \leq \frac{\mathbf{E}(e^{tX})}{e^{t\lambda}}$$ **basic Chernoff′s inequality**

Chebyshev's inequalities are used to show that values of a random variable lie close to its average with high probability. The bounds they provide are called also **concentration bounds**. Better bounds can usually be obtained using Chernoff bounds discussed in Chapter 5.

FLIPPING COINS EXAMPLES on CHEBYSHEV INEQUALITIES

Let X be a sum of n independent fair coins and let X_i be an indicator variable for the event that the i-th coin comes up heads.

FLIPPING COINS EXAMPLES on CHEBYSHEV INEQUALITIES

Let X be a sum of n independent fair coins and let X_i be an indicator variable for the event that the i-th coin comes up heads. Then $\mathbf{E}(X_i) = \frac{1}{2}$, $\mathbf{E}(X) = \frac{n}{2}$,

$\mathrm{Var}[X_i] = \frac{1}{4}$ and $\mathrm{Var}[X] = \sum \mathrm{Var}[X_i] = \frac{n}{4}$.

Chebyshev's inequality

$$\Pr[|X - \mathbf{E}(X)| \geq \lambda] \leq \frac{V(X)}{\lambda^2}$$

for $\lambda = \frac{n}{2}$ gives

$$Pr[X = n] \leq Pr[|X - n/2| \geq n/2] \leq \frac{n/4}{(n/2)^2} = \frac{1}{n}$$

THE INCLUSION-EXCLUSION PRINCIPLE

Let $A_1, A_2, \ldots, A_n$ be events – not necessarily disjoint. The **Inclusion-Exclusion principle**, that has also a variety of applications, states that

$$\begin{aligned} Pr\left[\bigcup_{i=1}^{n} A_i\right] &= \sum_{i=1}^{n} Pr(A_i) - \sum_{i<j} Pr(A_i \cap A_j) + \sum_{i<j<k} Pr(A_i \cap A_j \cap A_k) - \\ &- \ \ldots \ + (-1)^{k+1} \sum_{i_1<i_2<\ldots<i_k} Pr\left[\bigcap_{j=1}^{k} A_{i_j}\right] \ \ldots \ + \\ &+ (-1)^{n+1}\, Pr\left[\bigcap_{i=1}^{n} A_i\right] \end{aligned}$$

BONFERRONI'S INEQUALITIES

BONFERRONI'S INEQUALITIES

the following **Bonferroni's inequalities** follow from the Inclusion-exclusion principle:

For every odd $k \leq n$

$$Pr\left(\bigcup_{i=1}^{n} A_i\right) \leq \sum_{j=1}^{k} (-1)^{j+1} \sum_{i_1 < \ldots < i_j \leq n} Pr\left(\bigcap_{l=1}^{j} A_{i_l}\right)$$

For every even $k \leq n$

$$Pr\left(\bigcup_{i=1}^{n} A_i\right) \geq \sum_{j=1}^{k} (-1)^{j+1} \sum_{i_1 < \ldots < i_j \leq n} Pr\left(\bigcap_{l=1}^{j} A_{i_l}\right)$$

SPECIAL CASES of THE INCLUSION-EXCLUSION PRINCIPLE

SPECIAL CASES of THE INCLUSION-EXCLUSION PRINCIPLE

"Markov"-type inequality - Boole's inequality or Union bound

$$Pr\left(\bigcup_i A_i\right) \leq \sum_i Pr(A_i)$$

SPECIAL CASES of THE INCLUSION-EXCLUSION PRINCIPLE

"Markov"-type inequality - Boole's inequality or Union bound

$$Pr\left(\bigcup_i A_i\right) \leq \sum_i Pr(A_i)$$

"Chebyshev"-type inequality

$$Pr\left(\bigcup_i A_i\right) \geq \sum_i Pr(A_i) - \sum_{i<j} Pr(A_i \cap A_j)$$

SPECIAL CASES of THE INCLUSION-EXCLUSION PRINCIPLE

"Markov"-type inequality - Boole's inequality or Union bound

$$Pr\left(\bigcup_i A_i\right) \leq \sum_i Pr(A_i)$$

"Chebyshev"-type inequality

$$Pr\left(\bigcup_i A_i\right) \geq \sum_i Pr(A_i) - \sum_{i<j} Pr(A_i \cap A_j)$$

Another proof of Boole's inequality:

Let us define $B_i = A_i - \bigcup_{j=1}^{i-1} A_j$. Then $\bigcup A_i = \bigcup B_i$. Since B_i are disjoint and for each i we have $B_i \subset A_i$ we get

$$Pr[\bigcup A_i] = Pr[\bigcup B_i] = \sum Pr[B_i] \leq \sum Pr[A_i]$$

APPENDIX

APPENDIX

PUZZLE - HOMEWORK

Puzzle 1 Given a biased coin, how to use it to simulate an unbiased coin?

Puzzle 2 *n* people sit in a circle. Each person wears either red hat or a blue hat, chosen independently and uniformly at random. Each person can see the hats of all the other people, but not his/her hat. Based only upon what they see, each person votes on whether or not the total number of red hats is odd. Is there a scheme by which the outcome of the vote is correct with probability greater than 1/2.

MODERN (BAYESIAN) INTERPRETATION of BAYES RULE

Bayes rule for the process of learning from evidence has the form:

$$Pr[\varepsilon_1|\varepsilon] = \frac{Pr[\varepsilon_1 \cap \varepsilon]}{Pr[\varepsilon]} = \frac{Pr[\varepsilon|\varepsilon_1] \cdot Pr[\varepsilon_1]}{\sum_{i=1}^{k} Pr[\varepsilon|\varepsilon_i] \cdot Pr[\varepsilon_i]}.$$

In modern terms the last equation says that $Pr[\varepsilon_1|\varepsilon]$, the probability of a hypothesis ε_1 (given information ε), equals $\Pr(\varepsilon_1)$, our initial estimate of its probability, times $Pr[\varepsilon|\varepsilon_1]$, the probability of each new piece of information (under the hypothesis ε_1), divided by the sum of the probabilities of data in all possible hypothesis (ε_i).

EXAMPLE - DRUG TESTING

Suppose that a drug test will produce 99% true positive and 99% true negative results.

Suppose that 0.5% of people are drug users.

If the test of a user is positive, what is probability that such a user is a drug user?

SOLUTION

$$\Pr(\text{drg-us}|+) = \frac{\Pr(+|\text{drg-us})\Pr(\text{drg-us})}{\Pr(+|\text{drg-us})\Pr(\text{drg-us}) + \Pr(+|\text{no-drg-us})\Pr(\text{no-drg-us})}$$

$$\Pr(drg - us|+) = \frac{0.99 \times 0.005}{0.99 \times 0.005 + 0.01 \times 0.995} = \approx 33.2\%$$

BAYES' RULE INFORMALLY

Basically, Bayes' rule concerns of a broad and fundamental issue: how we analyze evidence and change our mind as we get new information, and make rational decision in the face of uncertainty.

Bayes' rule as one line theorem: by updating our initial belief about something with new objective information, we get a new and improved belief

BAYES' RULE STORY

- Reverend Thomas Bayes from England discovered the initial version of the "Bayes's law" around 1974, but soon stopped to believe in it.
- In behind were two philosophical questions
  - Can an effect determine its cause?
  - Can we determine the existence of God by observing nature?
- Bayes law was not written for long time as formula, only as the statement: **By updating our initial belief about something with objective new information, we can get a new and improved belief.**
- Bayes used a tricky thought experiment to demonstrate his law.
- Bayes' rule was later invented independently by Pierre Simon Laplace, perhaps the greatest scientist of 18th century, but at the end he also abounded it.
- Till the 20 century theoreticians considered Bayes rule as unscientific. Bayes rule had for centuries several proponents and many opponents in spite that it has turned out to be very useful in practice.
- Bayes rule was used to help to create rules of insurance industries, to develop strategy for artillery during the first and even Second World War (and also a great Russian mathematician Kolmogorov helped to develop it for this purpose).

- It was used much to decrypt ENIGMA codes during 2WW, due to Turing, and also to locate German submarines.

Part II

Basic Methods of design and Analysis of Randomized Algorithms

Chapter 4. BASIC TECHNIQUES for DESIGN and ANALYSIS

Chapter 4. BASIC TECHNIQUES for DESIGN and ANALYSIS

In this chapter we present a new way how to see randomized algorithms and several basic techniques how to design and analyse randomized algorithms:

Chapter 4. BASIC TECHNIQUES for DESIGN and ANALYSIS

In this chapter we present a new way how to see randomized algorithms and several basic techniques how to design and analyse randomized algorithms:

Especially we deal with:

Chapter 4. BASIC TECHNIQUES for DESIGN and ANALYSIS

In this chapter we present a new way how to see randomized algorithms and several basic techniques how to design and analyse randomized algorithms:

Especially we deal with:

- Application of linearity of expectations

Chapter 4. BASIC TECHNIQUES for DESIGN and ANALYSIS

In this chapter we present a new way how to see randomized algorithms and several basic techniques how to design and analyse randomized algorithms:

Especially we deal with:

- Application of linearity of expectations
- Game theory based lower bounds methods for randomized algorithms.

PROLOGUE

A way to see basics of deterministic, randomized and quantum computations and their differences.

MATHEMATICAL VIEWS of COMPUTATION 1/3

MATHEMATICAL VIEWS of COMPUTATION 1/3

Let us consider an n bits strings set $S \subset \{0,1\}^n$.

MATHEMATICAL VIEWS of COMPUTATION 1/3

Let us consider an n bits strings set $S \subset \{0,1\}^n$.

To describe a **deterministic computation** on S we need to specify:

MATHEMATICAL VIEWS of COMPUTATION 1/3

Let us consider an n bits strings set $S \subset \{0,1\}^n$.

To describe a **deterministic computation** on S we need to specify:

an initial state - by an n-bit string - say s_0

MATHEMATICAL VIEWS of COMPUTATION 1/3

Let us consider an n bits strings set $S \subset \{0,1\}^n$.

To describe a **deterministic computation** on S we need to specify:

an initial state - by an n-bit string - say s_0

and an **evolution** (computation) mapping $E : S \rightarrow S$ which can be described by a vector of the length 2^n, the elements and indices of which are n-bit strings.

MATHEMATICAL VIEWS of COMPUTATION 1/3

Let us consider an n bits strings set $S \subset \{0,1\}^n$.

To describe a **deterministic computation** on S we need to specify:

an initial state - by an n-bit string - say s_0

and an **evolution** (computation) mapping $E : S \rightarrow S$ which can be described by a vector of the length 2^n, the elements and indices of which are n-bit strings.

A **computation step** is then an application of the evolution mapping E to the current state represented by an n-bit string s.

MATHEMATICAL VIEWS of COMPUTATION 1/3

Let us consider an n bits strings set $S \subset \{0,1\}^n$.

To describe a **deterministic computation** on S we need to specify:

an initial state - by an n-bit string - say s_0

and an **evolution** (computation) mapping $E : S \to S$ which can be described by a vector of the length 2^n, the elements and indices of which are n-bit strings.

A **computation step** is then an application of the evolution mapping E to the current state represented by an n-bit string s.

However, for any at least a bit significant task, the number of bits needed to describe such an evolution mapping, $n2^n$, is much too big.

MATHEMATICAL VIEWS of COMPUTATION 1/3

Let us consider an n bits strings set $S \subset \{0,1\}^n$.

To describe a **deterministic computation** on S we need to specify:

an initial state - by an n-bit string - say s_0

and an **evolution** (computation) mapping $E : S \rightarrow S$ which can be described by a vector of the length 2^n, the elements and indices of which are n-bit strings.

A **computation step** is then an application of the evolution mapping E to the current state represented by an n-bit string s.

However, for any at least a bit significant task, the number of bits needed to describe such an evolution mapping, $n2^n$, is much too big. **The task of programming is then/therefore** to replace an application of such an enormously huge mapping by an application of a much shorter circuit/program.

MATHEMATICAL VIEWS of COMPUTATION 2/3

MATHEMATICAL VIEWS of COMPUTATION 2/3

To describe a **randomized computation** we need;

MATHEMATICAL VIEWS of COMPUTATION 2/3

To describe a **randomized computation** we need;

1:) to specify an **initial probability distribution** on all n-bit strings.

MATHEMATICAL VIEWS of COMPUTATION 2/3

To describe a **randomized computation** we need;

1:) to specify an **initial probability distribution** on all n-bit strings. That can be done by a vector of length 2^n, indexed by n-bit strings, the elements of which are non-negative numbers that sum up to 1.

MATHEMATICAL VIEWS of COMPUTATION 2/3

To describe a **randomized computation** we need;

1:) to specify an **initial probability distribution** on all n-bit strings. That can be done by a vector of length 2^n, indexed by n-bit strings, the elements of which are non-negative numbers that sum up to 1.

2:) to specify a **randomized evolution**, which has to be done, in case of a homogeneous evolution, by a $2^n \times 2^n$ matrix A of conditional probabilities for obtaining a new state/string from an old state/string.

MATHEMATICAL VIEWS of COMPUTATION 2/3

To describe a **randomized computation** we need;

1:) to specify an **initial probability distribution** on all n-bit strings. That can be done by a vector of length 2^n, indexed by n-bit strings, the elements of which are non-negative numbers that sum up to 1.

2:) to specify a **randomized evolution**, which has to be done, in case of a homogeneous evolution, by a $2^n \times 2^n$ matrix A of conditional probabilities for obtaining a new state/string from an old state/string.

The matrix A has to be **stochastic** - all columns have to sum up to one and $A[i, j]$ is a probability of going from a string representing j to a string representing i.

MATHEMATICAL VIEWS of COMPUTATION 2/3

To describe a **randomized computation** we need;

1:) to specify an **initial probability distribution** on all n-bit strings. That can be done by a vector of length 2^n, indexed by n-bit strings, the elements of which are non-negative numbers that sum up to 1.

2:) to specify a **randomized evolution**, which has to be done, in case of a homogeneous evolution, by a $2^n \times 2^n$ matrix A of conditional probabilities for obtaining a new state/string from an old state/string.

The matrix A has to be **stochastic** - all columns have to sum up to one and $A[i,j]$ is a probability of going from a string representing j to a string representing i.

To perform a computation step, one then needs to multiply by A the 2^n-elements vector specifying the current probability distribution on 2^n states.

MATHEMATICAL VIEWS of COMPUTATION 2/3

To describe a **randomized computation** we need;

1:) to specify an **initial probability distribution** on all n-bit strings. That can be done by a vector of length 2^n, indexed by n-bit strings, the elements of which are non-negative numbers that sum up to 1.

2:) to specify a **randomized evolution**, which has to be done, in case of a homogeneous evolution, by a $2^n \times 2^n$ matrix A of conditional probabilities for obtaining a new state/string from an old state/string.

The matrix A has to be **stochastic** - all columns have to sum up to one and $A[i,j]$ is a probability of going from a string representing j to a string representing i.

To perform a computation step, one then needs to multiply by A the 2^n-elements vector specifying the current probability distribution on 2^n states.

However, for any nontrivial problem the number 2^n is larger than the number of particles in the universe.

MATHEMATICAL VIEWS of COMPUTATION 2/3

To describe a **randomized computation** we need;

1:) to specify an **initial probability distribution** on all n-bit strings. That can be done by a vector of length 2^n, indexed by n-bit strings, the elements of which are non-negative numbers that sum up to 1.

2:) to specify a **randomized evolution**, which has to be done, in case of a homogeneous evolution, by a $2^n \times 2^n$ matrix A of conditional probabilities for obtaining a new state/string from an old state/string.

The matrix A has to be **stochastic** - all columns have to sum up to one and $A[i,j]$ is a probability of going from a string representing j to a string representing i.

To perform a computation step, one then needs to multiply by A the 2^n-elements vector specifying the current probability distribution on 2^n states.

However, for any nontrivial problem the number 2^n is larger than the number of particles in the universe. Therefore, **the task of programming is to design a small circuit/program** that can implement such a multiplication by a matrix of an enormous size.

MATHEMATICAL VIEWS of COMPUTATION 3/3

In case of **quantum computation** on n quantum bits:

---

[1] A matrix A is usually called unitary if its inverse matrix can be obtained from A by transposition around the main diagonal and replacement of each element by its complex conjugate.

MATHEMATICAL VIEWS of COMPUTATION 3/3

In case of **quantum computation** on n quantum bits:

1:) **Initial state** has to be given by an 2^n vector of complex numbers (probability amplitudes) the sum of the squares of which is one.

---

[1] A matrix A is usually called unitary if its inverse matrix can be obtained from A by transposition around the main diagonal and replacement of each element by its complex conjugate.

MATHEMATICAL VIEWS of COMPUTATION 3/3

In case of **quantum computation** on n quantum bits:

1:) **Initial state** has to be given by an 2^n vector of complex numbers (probability amplitudes) the sum of the squares of which is one.

2:) Homogeneous **quantum evolution** has to be described by an $2^n \times 2^n$ unitary matrix of complex numbers - at which inner products of any two different columns and any two different rows are 0.[1]

---

[1] A matrix A is usually called unitary if its inverse matrix can be obtained from A by transposition around the main diagonal and replacement of each element by its complex conjugate.

MATHEMATICAL VIEWS of COMPUTATION 3/3

In case of **quantum computation** on n quantum bits:

1:) **Initial state** has to be given by an 2^n vector of complex numbers (probability amplitudes) the sum of the squares of which is one.

2:) Homogeneous **quantum evolution** has to be described by an $2^n \times 2^n$ unitary matrix of complex numbers - at which inner products of any two different columns and any two different rows are 0.[1]

Concerning a **computation step**, this has to be again a multiplication of a vector of the probability amplitudes, representing the current state, by a very huge $2^n \times 2^n$ unitary matrix which has to be realized by a "small" quantum circuit (program).

---

[1] A matrix A is usually called unitary if its inverse matrix can be obtained from A by transposition around the main diagonal and replacement of each element by its complex conjugate.

LINEARITY OF EXPECTATIONS

LINEARITY OF EXPECTATIONS

A very simple, but very often very useful, fact is that for any random variables $X_1, X_2, \ldots$ it holds

$$\mathbf{E}[\sum_i X_i] = \sum_i \mathbf{E}[X_i].$$

LINEARITY OF EXPECTATIONS

A very simple, but very often very useful, fact is that for any random variables $X_1, X_2, \ldots$ it holds

$$\mathbf{E}[\sum_i X_i] = \sum_i \mathbf{E}[X_i].$$

even if X_i are dependent and dependencies among X_i's are very complex.

LINEARITY OF EXPECTATIONS

A very simple, but very often very useful, fact is that for any random variables $X_1, X_2, \ldots$ it holds

$$\mathbf{E}[\sum_i X_i] = \sum_i \mathbf{E}[X_i].$$

even if X_i are dependent and dependencies among X_i's are very complex.

Example: A ship arrives at a port, and all 40 sailors on board go ashore to have fun. At night, all sailors return to the ship and, being drunk, each chooses randomly a cabin to sleep in.

LINEARITY OF EXPECTATIONS

A very simple, but very often very useful, fact is that for any random variables $X_1, X_2, \ldots$ it holds

$$\mathbf{E}[\sum_i X_i] = \sum_i \mathbf{E}[X_i].$$

even if X_i are dependent and dependencies among X_i's are very complex.

Example: A ship arrives at a port, and all 40 sailors on board go ashore to have fun. At night, all sailors return to the ship and, being drunk, each chooses randomly a cabin to sleep in. Now comes the **question:** *What is the expected number of sailors sleeping in their own cabins?*

LINEARITY OF EXPECTATIONS

A very simple, but very often very useful, fact is that for any random variables $X_1, X_2, \ldots$ it holds

$$\mathbf{E}[\sum_i X_i] = \sum_i \mathbf{E}[X_i].$$

even if X_i are dependent and dependencies among X_i's are very complex.

Example: A ship arrives at a port, and all 40 sailors on board go ashore to have fun. At night, all sailors return to the ship and, being drunk, each chooses randomly a cabin to sleep in. Now comes the **question:** *What is the expected number of sailors sleeping in their own cabins?*

Solution: Let X_i be a random variable, so called *(indicator variable)*, which has value 1 if the i-th sailor chooses his own cabin, and 0 otherwise.

LINEARITY OF EXPECTATIONS

A very simple, but very often very useful, fact is that for any random variables $X_1, X_2, \ldots$ it holds

$$\mathbf{E}[\sum_i X_i] = \sum_i \mathbf{E}[X_i].$$

even if X_i are dependent and dependencies among X_i's are very complex.

Example: A ship arrives at a port, and all 40 sailors on board go ashore to have fun. At night, all sailors return to the ship and, being drunk, each chooses randomly a cabin to sleep in. Now comes the **question:** *What is the expected number of sailors sleeping in their own cabins?*

Solution: Let X_i be a random variable, so called *(indicator variable)*, which has value 1 if the i-th sailor chooses his own cabin, and 0 otherwise.

Expected number of sailors who get to their own cabin is

$$\mathbf{E}[\sum_{i=1}^{40} X_i] = \sum_{i=1}^{40} \mathbf{E}[X_i]$$

LINEARITY OF EXPECTATIONS

A very simple, but very often very useful, fact is that for any random variables $X_1, X_2, \ldots$ it holds

$$\mathbf{E}[\sum_i X_i] = \sum_i \mathbf{E}[X_i].$$

even if X_i are dependent and dependencies among X_i's are very complex.

Example: A ship arrives at a port, and all 40 sailors on board go ashore to have fun. At night, all sailors return to the ship and, being drunk, each chooses randomly a cabin to sleep in. Now comes the **question:** *What is the expected number of sailors sleeping in their own cabins?*

Solution: Let X_i be a random variable, so called *(indicator variable)*, which has value 1 if the i-th sailor chooses his own cabin, and 0 otherwise.

Expected number of sailors who get to their own cabin is

$$\mathbf{E}[\sum_{i=1}^{40} X_i] = \sum_{i=1}^{40} \mathbf{E}[X_i]$$

Since cabins are chosen randomly $\mathbf{E}[X_i] =$

LINEARITY OF EXPECTATIONS

A very simple, but very often very useful, fact is that for any random variables $X_1, X_2, \ldots$ it holds

$$\mathbf{E}[\sum_i X_i] = \sum_i \mathbf{E}[X_i].$$

even if X_i are dependent and dependencies among X_i's are very complex.

Example: A ship arrives at a port, and all 40 sailors on board go ashore to have fun. At night, all sailors return to the ship and, being drunk, each chooses randomly a cabin to sleep in. Now comes the **question:** *What is the expected number of sailors sleeping in their own cabins?*

Solution: Let X_i be a random variable, so called *(indicator variable)*, which has value 1 if the i-th sailor chooses his own cabin, and 0 otherwise.

Expected number of sailors who get to their own cabin is

$$\mathbf{E}[\sum_{i=1}^{40} X_i] = \sum_{i=1}^{40} \mathbf{E}[X_i]$$

Since cabins are chosen randomly $\mathbf{E}[X_i] = \frac{1}{40}$

LINEARITY OF EXPECTATIONS

A very simple, but very often very useful, fact is that for any random variables $X_1, X_2, \ldots$ it holds

$$\mathbf{E}[\sum_i X_i] = \sum_i \mathbf{E}[X_i].$$

even if X_i are dependent and dependencies among X_i's are very complex.

Example: A ship arrives at a port, and all 40 sailors on board go ashore to have fun. At night, all sailors return to the ship and, being drunk, each chooses randomly a cabin to sleep in. Now comes the **question:** *What is the expected number of sailors sleeping in their own cabins?*

Solution: Let X_i be a random variable, so called *(indicator variable)*, which has value 1 if the i-th sailor chooses his own cabin, and 0 otherwise.

Expected number of sailors who get to their own cabin is

$$\mathbf{E}[\sum_{i=1}^{40} X_i] = \sum_{i=1}^{40} \mathbf{E}[X_i]$$

Since cabins are chosen randomly $\mathbf{E}[X_i] = \frac{1}{40}$ and

LINEARITY OF EXPECTATIONS

A very simple, but very often very useful, fact is that for any random variables $X_1, X_2, \ldots$ it holds

$$\mathbf{E}[\sum_i X_i] = \sum_i \mathbf{E}[X_i].$$

even if X_i are dependent and dependencies among X_i's are very complex.

Example: A ship arrives at a port, and all 40 sailors on board go ashore to have fun. At night, all sailors return to the ship and, being drunk, each chooses randomly a cabin to sleep in. Now comes the **question:** *What is the expected number of sailors sleeping in their own cabins?*

Solution: Let X_i be a random variable, so called *(indicator variable)*, which has value 1 if the i-th sailor chooses his own cabin, and 0 otherwise.

Expected number of sailors who get to their own cabin is

$$\mathbf{E}[\sum_{i=1}^{40} X_i] = \sum_{i=1}^{40} \mathbf{E}[X_i]$$

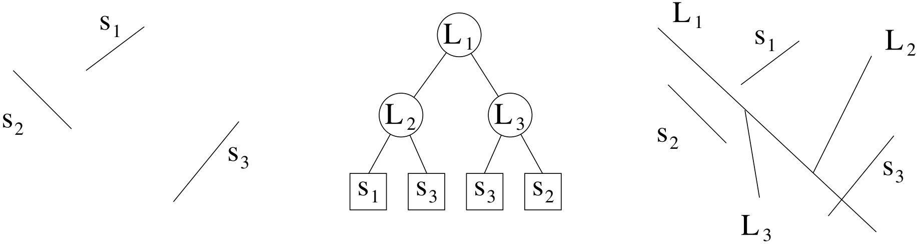
Since cabins are chosen randomly $\mathbf{E}[X_i] = \frac{1}{40}$ and $\mathbf{E}[\sum_{i=1}^{40} X_i] = 40.\frac{1}{40} = 1$.

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 1/3

Problem Given a set $S = \{s_1, \ldots, s_n\}$ of non-intersecting line segments, find a partition of the plane such that every region will contain at most one line segment (or at most a part of a line segment).

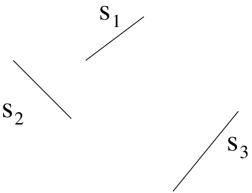
EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 1/3

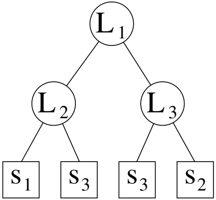
Problem Given a set $S = \{s_1, \ldots, s_n\}$ of non-intersecting line segments, find a partition of the plane such that every region will contain at most one line segment (or at most a part of a line segment).

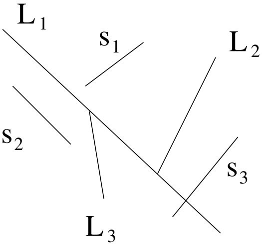


EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 1/3

Problem Given a set $S = \{s_1, \ldots, s_n\}$ of non-intersecting line segments, find a partition of the plane such that every region will contain at most one line segment (or at most a part of a line segment).



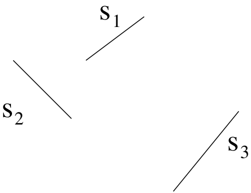


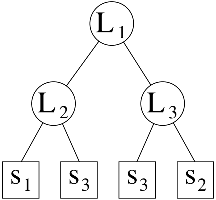


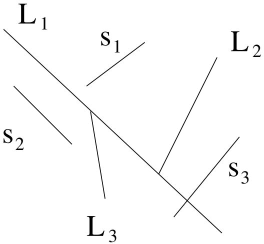
A (binary) partition will be described by a binary tree + additional information (about nodes).

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 1/3

Problem Given a set $S = \{s_1, \ldots, s_n\}$ of non-intersecting line segments, find a partition of the plane such that every region will contain at most one line segment (or at most a part of a line segment).



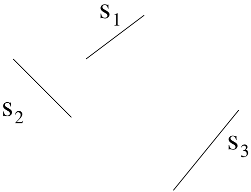


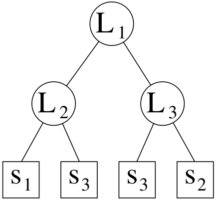


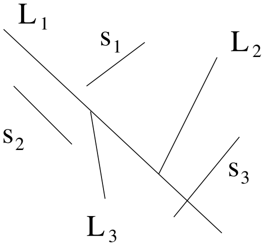
A (binary) partition will be described by a binary tree + additional information (about nodes). With each node v a region r_v of the plane will be associated (the whole plane will be represented by the root) and also a line L_v intersecting r_v.

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 1/3

Problem Given a set $S = \{s_1, \ldots, s_n\}$ of non-intersecting line segments, find a partition of the plane such that every region will contain at most one line segment (or at most a part of a line segment).







A (binary) partition will be described by a binary tree + additional information (about nodes). With each node v a region r_v of the plane will be associated (the whole plane will be represented by the root) and also a line L_v intersecting r_v.

Each line L_v will partition the region r_v into two regions $r_{l,v}$ and $r_{r,v}$ which correspond to two children of v - to the left and right one.

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 2/3

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 2/3

Notation: $l(s_i)$ will denote a **line-extension of the segment** s_i.

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 2/3

Notation: $l(s_i)$ will denote a **line-extension of the segment** s_i.
autopartitions will use only line-extensions of given segments.

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 2/3

Notation: $l(s_i)$ will denote a **line-extension of the segment s_i**.

autopartitions will use only line-extensions of given segments.

Algorithm RandAuto:

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 2/3

Notation: $l(s_i)$ will denote a **line-extension of the segment** s_i.
autopartitions will use only line-extensions of given segments.
Algorithm RandAuto:

Input: A set $S = \{s_1, \ldots, s_n\}$ of non-intersecting line segments.

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 2/3

Notation: $l(s_i)$ will denote a **line-extension of the segment** s_i.
autopartitions will use only line-extensions of given segments.
Algorithm RandAuto:

Input: A set $S = \{s_1, \ldots, s_n\}$ of non-intersecting line segments.
Output: A binary autopartition P_Π of S.

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 2/3

Notation: $l(s_i)$ will denote a **line-extension of the segment s_i**.

autopartitions will use only line-extensions of given segments.

Algorithm RandAuto:

```
    **Input:** A set $S = \{s_1, \ldots, s_n\}$ of non-intersecting line segments.
    *Output:* A binary autopartition $P_\Pi$ of $S$.
    *1:* Pick a permutation $\Pi$ of $\{1, \ldots, n\}$ uniformly and randomly.
```

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 2/3

Notation: $l(s_i)$ will denote a **line-extension of the segment** s_i.
autopartitions will use only line-extensions of given segments.
Algorithm RandAuto:

```
Input: A set $S = \{s_1, \ldots, s_n\}$ of non-intersecting line segments.
Output: A binary autopartition $P_\Pi$ of $S$.
1: Pick a permutation $\Pi$ of $\{1, \ldots, n\}$ uniformly and randomly.
2: While there is a region $R$ that contains more than one segment, choose one of them randomly and cut it with $l(s_i)$ where $i$ is the first element in the ordering induced by $\Pi$ such that $l(s_i)$ cuts the region $R$.
```

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 2/3

Notation: $l(s_i)$ will denote a **line-extension of the segment** s_i.
autopartitions will use only line-extensions of given segments.
Algorithm RandAuto:

Input: A set $S = \{s_1, \ldots, s_n\}$ of non-intersecting line segments.

Output: A binary autopartition P_Π of S.

1: Pick a permutation Π of $\{1, \ldots, n\}$ uniformly and randomly.

2: **While** there is a region R that contains more than one segment, choose one of them randomly and cut it with $l(s_i)$ where i is the first element in the ordering induced by Π such that $l(s_i)$ cuts the region R.

Theorem: The expected size of the autopartition P_Π of S, produced by the above RandAuto algorithm is $\theta(n \ln n)$.

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 2/3

Notation: $l(s_i)$ will denote a **line-extension of the segment** s_i.
autopartitions will use only line-extensions of given segments.
Algorithm RandAuto:

Input: A set $S = \{s_1, \ldots, s_n\}$ of non-intersecting line segments.

Output: A binary autopartition P_Π of S.

1: Pick a permutation Π of $\{1, \ldots, n\}$ uniformly and randomly.

2: **While** there is a region R that contains more than one segment, choose one of them randomly and cut it with $l(s_i)$ where i is the first element in the ordering induced by Π such that $l(s_i)$ cuts the region R.

Theorem: The expected size of the autopartition P_Π of S, produced by the above RandAuto algorithm is $\theta(n \ln n)$.

Proof: **Notation** (for line segments u, v).

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 2/3

Notation: $l(s_i)$ will denote a **line-extension of the segment** s_i.
autopartitions will use only line-extensions of given segments.
Algorithm RandAuto:

Input: A set $S = \{s_1, \ldots, s_n\}$ of non-intersecting line segments.
Output: A binary autopartition P_Π of S.
1: Pick a permutation Π of $\{1, \ldots, n\}$ uniformly and randomly.
2: **While** there is a region R that contains more than one segment, choose one of them randomly and cut it with $l(s_i)$ where i is the first element in the ordering induced by Π such that $l(s_i)$ cuts the region R.

Theorem: The expected size of the autopartition P_Π of S, produced by the above RandAuto algorithm is $\theta(n \ln n)$.

Proof: **Notation** (for line segments u, v).

$$index(u,v) = \begin{array}{lll} i & \text{if} & l(u) \text{ intersects } i-1 \text{ segments before hitting } v; \\ \infty & & \text{if } l(u) \text{ does not hit } v. \end{array}$$

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 2/3

Notation: $l(s_i)$ will denote a **line-extension of the segment s_i**.
autopartitions will use only line-extensions of given segments.
Algorithm RandAuto:

```
Input: A set $S = \{s_1, \ldots, s_n\}$ of non-intersecting line segments.
Output: A binary autopartition $P_\Pi$ of $S$.
1: Pick a permutation $\Pi$ of $\{1, \ldots, n\}$ uniformly and randomly.
2: While there is a region $R$ that contains more than one segment, choose one of them randomly and cut it with $l(s_i)$ where $i$ is the first element in the ordering induced by $\Pi$ such that $l(s_i)$ cuts the region $R$.
```

Theorem: The expected size of the autopartition P_Π of S, produced by the above RandAuto algorithm is $\theta(n \ln n)$.

Proof: **Notation** (for line segments u, v).

$$index(u, v) = \begin{array}{ll} i \quad \text{if} & l(u) \text{ intersects } i-1 \text{ segments before hitting } v; \\ \infty & \text{if } l(u) \text{ does not hit } v. \end{array}$$

$u \dashv v$ will be an event that $l(u)$ cuts v in the constructed (autopartition) tree.

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 3/3

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 3/3

Probability: Let u and v be segments, $index(u, v) = i$ and let $u_1, \ldots, u_{i-1}$ be segments the line $l(u)$ intersects before hitting v.

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 3/3

Probability: Let u and v be segments, $index(u, v) = i$ and let $u_1, \ldots, u_{i-1}$ be segments the line $l(u)$ intersects before hitting v.
The event $u \dashv v$ happens, during an execution of RandPart, only if u occurs before any of $\{u_1, \ldots, u_{i-1}, v\}$ in the permutation Π.

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 3/3

Probability: Let u and v be segments, $index(u, v) = i$ and let $u_1, \ldots, u_{i-1}$ be segments the line $l(u)$ intersects before hitting v.
The event $u \dashv v$ happens, during an execution of RandPart, only if u occurs before any of $\{u_1, \ldots, u_{i-1}, v\}$ in the permutation Π.Therefore the probability that event $u \dashv v$ happens is $\frac{1}{i+1} = \frac{1}{\text{index}(u,v)+1}$.

Notation: Let $C_{u,v}$ be the indicator variable that has value 1 if $u \dashv v$ and 0 otherwise.

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 3/3

Probability: Let u and v be segments, $index(u, v) = i$ and let $u_1, \ldots, u_{i-1}$ be segments the line $l(u)$ intersects before hitting v.

The event $u \dashv v$ happens, during an execution of RandPart, only if u occurs before any of $\{u_1, \ldots, u_{i-1}, v\}$ in the permutation Π. Therefore the probability that event $u \dashv v$ happens is $\frac{1}{i+1} = \frac{1}{\text{index}(u,v)+1}$.

Notation: Let $C_{u,v}$ be the indicator variable that has value 1 if $u \dashv v$ and 0 otherwise.

$$\mathbf{E}[C_{u,v}] = Pr[u \dashv v] = \frac{1}{index(u, v) + 1}.$$

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 3/3

Probability: Let u and v be segments, $index(u, v) = i$ and let $u_1, \ldots, u_{i-1}$ be segments the line $l(u)$ intersects before hitting v.

The event $u \dashv v$ happens, during an execution of RandPart, only if u occurs before any of $\{u_1, \ldots, u_{i-1}, v\}$ in the permutation Π.Therefore the probability that event $u \dashv v$ happens is $\frac{1}{i+1} = \frac{1}{\text{index}(u,v)+1}$.

Notation: Let $C_{u,v}$ be the indicator variable that has value 1 if $u \dashv v$ and 0 otherwise.

$$\mathbf{E}[C_{u,v}] = Pr[u \dashv v] = \frac{1}{index(u, v) + 1}.$$

Clearly, the size of the created partition P_Π equals n plus the number of intersections due to cuts.

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 3/3

Probability: Let u and v be segments, $index(u, v) = i$ and let $u_1, \ldots, u_{i-1}$ be segments the line $l(u)$ intersects before hitting v.
The event $u \dashv v$ happens, during an execution of RandPart, only if u occurs before any of $\{u_1, \ldots, u_{i-1}, v\}$ in the permutation Π.Therefore the probability that event $u \dashv v$ happens is $\frac{1}{i+1} = \frac{1}{\text{index}(u,v)+1}$.

Notation: Let $C_{u,v}$ be the indicator variable that has value 1 if $u \dashv v$ and 0 otherwise.

$$\mathbf{E}[C_{u,v}] = Pr[u \dashv v] = \frac{1}{index(u, v) + 1}.$$

Clearly, the size of the created partition P_Π equals n plus the number of intersections due to cuts. Its expectation value is therefore

$$n + E[\sum_u \sum_{v \neq u} C_{u,v}] = n + \sum_u \sum_{v \neq u} Pr[u \dashv v] = n + \sum_u \sum_{v \neq u} \frac{1}{index(u, v) + 1}.$$

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 3/3

Probability: Let u and v be segments, $index(u,v) = i$ and let $u_1, \ldots, u_{i-1}$ be segments the line $l(u)$ intersects before hitting v.

The event $u \dashv v$ happens, during an execution of RandPart, only if u occurs before any of $\{u_1, \ldots, u_{i-1}, v\}$ in the permutation Π.Therefore the probability that event $u \dashv v$ happens is $\frac{1}{i+1} = \frac{1}{\mathrm{index}(u,v)+1}$.

Notation: Let $C_{u,v}$ be the indicator variable that has value 1 if $u \dashv v$ and 0 otherwise.

$$\mathbf{E}[C_{u,v}] = Pr[u \dashv v] = \frac{1}{index(u,v)+1}.$$

Clearly, the size of the created partition P_Π equals n plus the number of intersections due to cuts. Its expectation value is therefore

$$n + E[\sum_u \sum_{v \neq u} C_{u,v}] = n + \sum_u \sum_{v \neq u} Pr[u \dashv v] = n + \sum_u \sum_{v \neq u} \frac{1}{index(u,v)+1}.$$

For any line segment u and integer i there are at most two v, w such that $index(u,v) = index(u,w) = i$.

EXAMPLE - BINARY PARTITION of a SET of LINE SEGMENTS 3/3

Probability: Let u and v be segments, $index(u, v) = i$ and let $u_1, \ldots, u_{i-1}$ be segments the line $l(u)$ intersects before hitting v.
The event $u \dashv v$ happens, during an execution of RandPart, only if u occurs before any of $\{u_1, \ldots, u_{i-1}, v\}$ in the permutation Π.Therefore the probability that event $u \dashv v$ happens is $\frac{1}{i+1} = \frac{1}{\text{index}(u,v)+1}$.

Notation: Let $C_{u,v}$ be the indicator variable that has value 1 if $u \dashv v$ and 0 otherwise.

$$\mathbf{E}[C_{u,v}] = Pr[u \dashv v] = \frac{1}{index(u, v) + 1}.$$

Clearly, the size of the created partition P_Π equals n plus the number of intersections due to cuts. Its expectation value is therefore

$$n + E[\sum_u \sum_{v \neq u} C_{u,v}] = n + \sum_u \sum_{v \neq u} Pr[u \dashv v] = n + \sum_u \sum_{v \neq u} \frac{1}{index(u, v) + 1}.$$

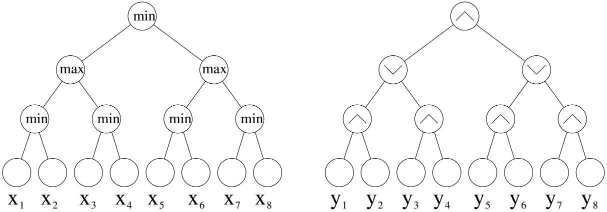
For any line segment u and integer i there are at most two v, w such that $index(u, v) = index(u, w) = i$.Hence $\sum_{v \neq u} \frac{1}{index(u,v)+1} \leq \sum_{i=1}^{n-1} \frac{2}{i+1}$ and therefore $n + \mathbf{E}[\sum_u \sum_{v \neq u} C_{u,v}] \leq n + \sum_u \sum_{i=1}^{n-1} \frac{2}{i+1} \leq n + 2nH_n$.

GAME TREE EVALUATION - I.

Game trees

GAME TREE EVALUATION - I.

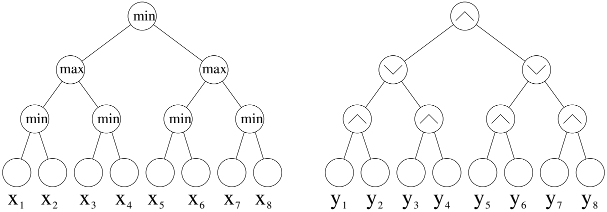
Game trees



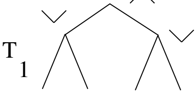
Game trees are trees with operations **max** and **min** alternating in internal nodes and values assigned to their leaves.

GAME TREE EVALUATION - I.

Game trees



Game trees are trees with operations **max** and **min** alternating in internal nodes and values assigned to their leaves. In case all such values are Boolean - **0** or **1** Boolean operation **OR** and **AND** are considered instead of **max** and **min**.

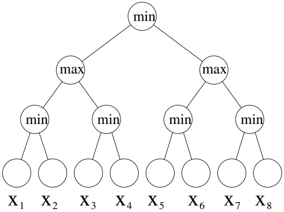


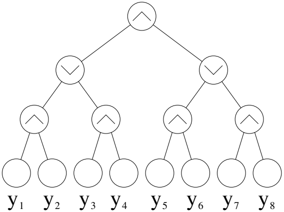
T_k – binary game tree of depth $2k$.

Goal is to evaluate the tree - the root.

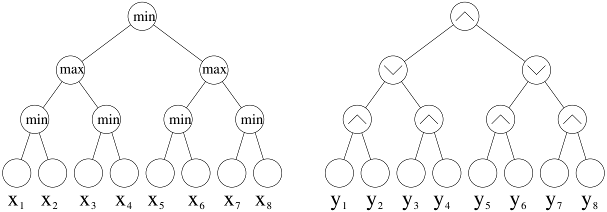
GAME TREE EVALUATION - II.

GAME TREE EVALUATION - II.



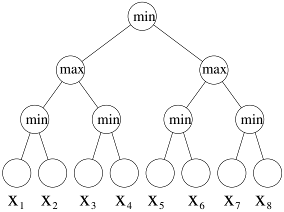


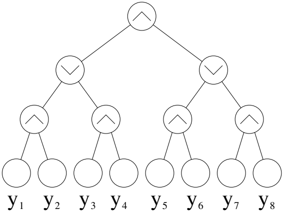
GAME TREE EVALUATION - II.



Evaluation of game trees plays a crucial role in AI, in various game playing programs.

GAME TREE EVALUATION - II.

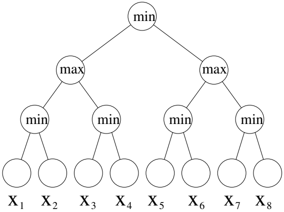


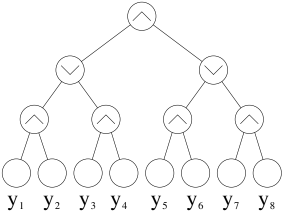


Evaluation of game trees plays a crucial role in AI, in various game playing programs.

Assumption: An evaluation algorithm chooses at each step (somehow) a leaf, reads its value and performs all evaluations of internal nodes it can perform.

GAME TREE EVALUATION - II.

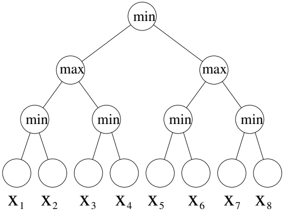


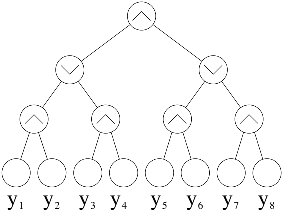


Evaluation of game trees plays a crucial role in AI, in various game playing programs.

Assumption: An evaluation algorithm chooses at each step (somehow) a leaf, reads its value and performs all evaluations of internal nodes it can perform. Cost of an evaluation algorithm is the number of leaves inspected.

GAME TREE EVALUATION - II.

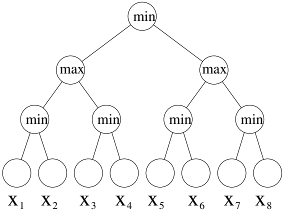


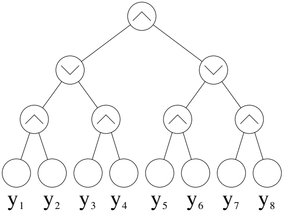


Evaluation of game trees plays a crucial role in AI, in various game playing programs.

Assumption: An evaluation algorithm chooses at each step (somehow) a leaf, reads its value and performs all evaluations of internal nodes it can perform. Cost of an evaluation algorithm is the number of leaves inspected. Determine the total number of such steps needed.

GAME TREE EVALUATION - II.





Evaluation of game trees plays a crucial role in AI, in various game playing programs.

Assumption: An evaluation algorithm chooses at each step (somehow) a leaf, reads its value and performs all evaluations of internal nodes it can perform. Cost of an evaluation algorithm is the number of leaves inspected. Determine the total number of such steps needed.

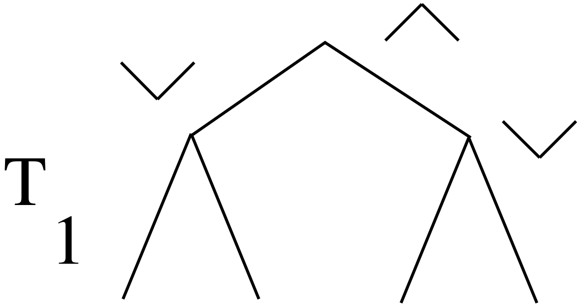
WORST CASE COMPLEXITY

WORST CASE COMPLEXITY

T_k – will denote the binary game tree of depth $2k$.

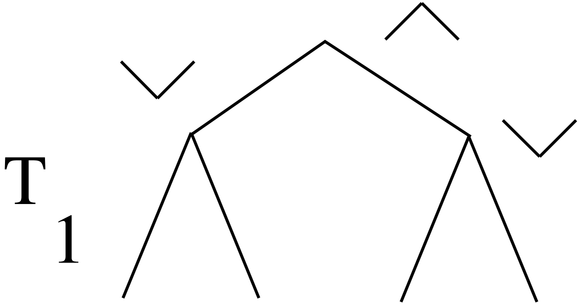
WORST CASE COMPLEXITY

T_k – will denote the binary game tree of depth $2k$.



WORST CASE COMPLEXITY

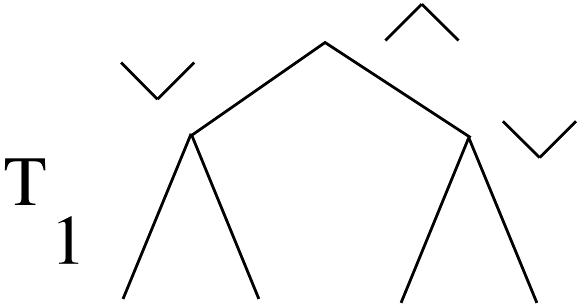
T_k – will denote the binary game tree of depth $2k$.



Every deterministic algorithm can be forced to inspect all leaves.

WORST CASE COMPLEXITY

T_k – will denote the binary game tree of depth $2k$.



Every deterministic algorithm can be forced to inspect all leaves. The worst-case complexity of a deterministic algorithm is therefore:

$$n = 4^k = 2^{2k}.$$

A RANDOMIZED ALGORITHM - BASIC IDEA:

A RANDOMIZED ALGORITHM - BASIC IDEA:

To evaluate an AND-node v, the algorithm chooses randomly one of its children and evaluates it.

A RANDOMIZED ALGORITHM - BASIC IDEA:

To evaluate an AND-node v, the algorithm chooses randomly one of its children and evaluates it.

If 1 is returned,

A RANDOMIZED ALGORITHM - BASIC IDEA:

To evaluate an AND-node v, the algorithm chooses randomly one of its children and evaluates it.

If 1 is returned, algorithm proceeds to evaluate other children subtree and returns as the value of v the value of that subtree.

A RANDOMIZED ALGORITHM - BASIC IDEA:

To evaluate an AND-node v, the algorithm chooses randomly one of its children and evaluates it.

If 1 is returned, algorithm proceeds to evaluate other children subtree and returns as the value of v the value of that subtree. If 0 is returned,

A RANDOMIZED ALGORITHM - BASIC IDEA:

To evaluate an AND-node v, the algorithm chooses randomly one of its children and evaluates it.

If 1 is returned, algorithm proceeds to evaluate other children subtree and returns as the value of v the value of that subtree. If 0 is returned, algorithm returns immediately 0 for v (without evaluating other subtree).

A RANDOMIZED ALGORITHM - BASIC IDEA:

To evaluate an AND-node v, the algorithm chooses randomly one of its children and evaluates it.

If 1 is returned, algorithm proceeds to evaluate other children subtree and returns as the value of v the value of that subtree. If 0 is returned, algorithm returns immediately 0 for v (without evaluating other subtree).

To evaluate an OR-node v, algorithm chooses randomly one of its children and evaluates it.

A RANDOMIZED ALGORITHM - BASIC IDEA:

To evaluate an AND-node v, the algorithm chooses randomly one of its children and evaluates it.

If 1 is returned, algorithm proceeds to evaluate other children subtree and returns as the value of v the value of that subtree. If 0 is returned, algorithm returns immediately 0 for v (without evaluating other subtree).

To evaluate an OR-node v, algorithm chooses randomly one of its children and evaluates it.

If 0 is returned, algorithm proceeds to evaluate other subtree and returns as the value of v the value of the subtree.

A RANDOMIZED ALGORITHM - BASIC IDEA:

To evaluate an AND-node v, the algorithm chooses randomly one of its children and evaluates it.

If 1 is returned, algorithm proceeds to evaluate other children subtree and returns as the value of v the value of that subtree. If 0 is returned, algorithm returns immediately 0 for v (without evaluating other subtree).

To evaluate an OR-node v, algorithm chooses randomly one of its children and evaluates it.

If 0 is returned, algorithm proceeds to evaluate other subtree and returns as the value of v the value of the subtree. If 1 is returned, the algorithm returns 1 for v.

RANDOMIZED ALGORITHMS - SUMMARY of THE BASIC IDEA

Start at the root and in order to evaluate a node evaluate (recursively) a random child of the current node.

RANDOMIZED ALGORITHMS - SUMMARY of THE BASIC IDEA

Start at the root and in order to evaluate a node evaluate (recursively) a random child of the current node.

If this does not determine the value of the current node, evaluate the node of other child.

Theorem: Given any instance of T_k, the expected number of steps for the above randomized algorithm is at most 3^k.

Theorem: Given any instance of T_k, the expected number of steps for the above randomized algorithm is at most 3^k.

Proof by induction:
Base step: Case $k = 1$ easy - verify by computations for all cases.

Theorem: Given any instance of T_k, the expected number of steps for the above randomized algorithm is at most 3^k.

Proof by induction:
Base step: Case $k = 1$ easy - verify by computations for all cases.
Inductive step: Assume that the expected cost of the evaluation of any instance of T_{k-1} is at most 3^{k-1}.

Theorem: Given any instance of T_k, the expected number of steps for the above randomized algorithm is at most 3^k.

Proof by induction:
Base step: Case $k = 1$ easy - verify by computations for all cases.
Inductive step: Assume that the expected cost of the evaluation of any instance of T_{k-1} is at most 3^{k-1}.

Consider an OR-node tree T with both children being T_{k-1}-trees.

Theorem: Given any instance of T_k, the expected number of steps for the above randomized algorithm is at most 3^k.

Proof by induction:
Base step: Case $k = 1$ easy - verify by computations for all cases.
Inductive step: Assume that the expected cost of the evaluation of any instance of T_{k-1} is at most 3^{k-1}.

Consider an OR-node tree T with both children being T_{k-1}-trees.
If the root of T were to return 1,

Theorem: Given any instance of T_k, the expected number of steps for the above randomized algorithm is at most 3^k.

Proof by induction:
Base step: Case $k = 1$ easy - verify by computations for all cases.
Inductive step: Assume that the expected cost of the evaluation of any instance of T_{k-1} is at most 3^{k-1}.

Consider an OR-node tree T with both children being T_{k-1}-trees.
If the root of T were to return 1, at least one of its T_{k-1}-subtrees has to return 1.

Theorem: Given any instance of T_k, the expected number of steps for the above randomized algorithm is at most 3^k.

Proof by induction:
Base step: Case $k = 1$ easy - verify by computations for all cases.
Inductive step: Assume that the expected cost of the evaluation of any instance of T_{k-1} is at most 3^{k-1}.

Consider an OR-node tree T with both children being T_{k-1}-trees.
If the root of T were to return 1, at least one of its T_{k-1}-subtrees has to return 1. With probability $\frac{1}{2}$ this child is chosen first, given in average at most 3^{k-1} leaf-evaluations.

Theorem: Given any instance of T_k, the expected number of steps for the above randomized algorithm is at most 3^k.

Proof by induction:
Base step: Case $k = 1$ easy - verify by computations for all cases.
Inductive step: Assume that the expected cost of the evaluation of any instance of T_{k-1} is at most 3^{k-1}.

Consider an OR-node tree T with both children being T_{k-1}-trees.
If the root of T were to return 1, at least one of its T_{k-1}-subtrees has to return 1. With probability $\frac{1}{2}$ this child is chosen first, given in average at most 3^{k-1} leaf-evaluations. With probability $\frac{1}{2}$ both subtrees are to be evaluated.

Theorem: Given any instance of T_k, the expected number of steps for the above randomized algorithm is at most 3^k.

Proof by induction:
Base step: Case $k = 1$ easy - verify by computations for all cases.
Inductive step: Assume that the expected cost of the evaluation of any instance of T_{k-1} is at most 3^{k-1}.

Consider an OR-node tree T with both children being T_{k-1}-trees.
If the root of T were to return 1, at least one of its T_{k-1}-subtrees has to return 1. With probability $\frac{1}{2}$ this child is chosen first, given in average at most 3^{k-1} leaf-evaluations. With probability $\frac{1}{2}$ both subtrees are to be evaluated. The expected cost of determining the value of T is therefore:

Theorem: Given any instance of T_k, the expected number of steps for the above randomized algorithm is at most 3^k.

Proof by induction:
Base step: Case $k = 1$ easy - verify by computations for all cases.
Inductive step: Assume that the expected cost of the evaluation of any instance of T_{k-1} is at most 3^{k-1}.

Consider an OR-node tree T with both children being T_{k-1}-trees.
If the root of T were to return 1, at least one of its T_{k-1}-subtrees has to return 1. With probability $\frac{1}{2}$ this child is chosen first, given in average at most 3^{k-1} leaf-evaluations. With probability $\frac{1}{2}$ both subtrees are to be evaluated. The expected cost of determining the value of T is therefore:

$$\frac{1}{2} \times 3^{k-1} + \frac{1}{2} \times 2 \times 3^{k-1} = \frac{1}{2} \times 3^k = \frac{3}{2} \times 3^{k-1}.$$

If the root of T were to return 0 both subtrees have to be evaluated, giving the cost $2 \times 3^{k-1}$.

If the root of T were to return 0 both subtrees have to be evaluated, giving the cost $2 \times 3^{k-1}$.

Consider now the root of T_k.

If the root of T were to return 0 both subtrees have to be evaluated, giving the cost $2 \times 3^{k-1}$.

Consider now the root of T_k.

If the root evaluates to 1, both of its OR-subtrees have to evaluate to 1.

If the root of T were to return 0 both subtrees have to be evaluated, giving the cost $2 \times 3^{k-1}$.

Consider now the root of T_k.

If the root evaluates to 1, both of its OR-subtrees have to evaluate to 1. The expected cost is therefore

$$2 \times \frac{3}{2} \times 3^{k-1} = 3^k.$$

If the root of T were to return 0 both subtrees have to be evaluated, giving the cost $2 \times 3^{k-1}$.

Consider now the root of T_k.

If the root evaluates to 1, both of its OR-subtrees have to evaluate to 1. The expected cost is therefore

$$2 \times \frac{3}{2} \times 3^{k-1} = 3^k.$$

If the root evaluates to 0, at least one of the subtrees evaluates to 0. The expected cost is therefore

$$\frac{1}{2} \times 2 \times 2 \times 3^{k-1} + \frac{1}{2} \times \frac{3}{2} \times 3^{k-1} \leq 3^k = n^{\lg_4 3} = n^{0.793}.$$

If the root of T were to return 0 both subtrees have to be evaluated, giving the cost $2 \times 3^{k-1}$.

Consider now the root of T_k.

If the root evaluates to 1, both of its OR-subtrees have to evaluate to 1. The expected cost is therefore

$$2 \times \frac{3}{2} \times 3^{k-1} = 3^k.$$

If the root evaluates to 0, at least one of the subtrees evaluates to 0. The expected cost is therefore

$$\frac{1}{2} \times 2 \times 2 \times 3^{k-1} + \frac{1}{2} \times \frac{3}{2} \times 3^{k-1} \leq 3^k = n^{\lg_4 3} = n^{0.793}.$$

Our algorithm is therefore a *Las Vegas algorithm*.

If the root of T were to return 0 both subtrees have to be evaluated, giving the cost $2 \times 3^{k-1}$.

Consider now the root of T_k.

If the root evaluates to 1, both of its OR-subtrees have to evaluate to 1. The expected cost is therefore

$$2 \times \frac{3}{2} \times 3^{k-1} = 3^k.$$

If the root evaluates to 0, at least one of the subtrees evaluates to 0. The expected cost is therefore

$$\frac{1}{2} \times 2 \times 2 \times 3^{k-1} + \frac{1}{2} \times \frac{3}{2} \times 3^{k-1} \leq 3^k = n^{\lg_4 3} = n^{0.793}.$$

Our algorithm is therefore a *Las Vegas algorithm*. Its running time (number of leaves evaluations) is: $n^{0.793}$.

CLASSICAL GAMES THEORY

CLASSICAL GAMES THEORY
BRIEFLY

BASIC CONCEPTS of CLASSICAL GAME THEORY

We will consider **games with two players**, Alice and Bob.

BASIC CONCEPTS of CLASSICAL GAME THEORY

We will consider **games with two players**, Alice and Bob. X and Y will be nonempty sets of their game **(pure) strategies** -X of Alice, Y of Bob.

BASIC CONCEPTS of CLASSICAL GAME THEORY

We will consider **games with two players**, Alice and Bob. X and Y will be nonempty sets of their game **(pure) strategies** -X of Alice, Y of Bob. Mappings $p_X : X \times Y \rightarrow \mathbf{R}$ and $p_Y : X \times Y \rightarrow \mathbf{R}$ will be called **payoff functions** of Alice and Bob.

BASIC CONCEPTS of CLASSICAL GAME THEORY

We will consider **games with two players**, Alice and Bob. X and Y will be nonempty sets of their game **(pure) strategies** -X of Alice, Y of Bob. Mappings $p_X : X \times Y \rightarrow \mathbf{R}$ and $p_Y : X \times Y \rightarrow \mathbf{R}$ will be called **payoff functions** of Alice and Bob. The quadruple (X, Y, p_X, p_Y) will be called a (mathematical) **game**.

BASIC CONCEPTS of CLASSICAL GAME THEORY

We will consider **games with two players**, Alice and Bob. X and Y will be nonempty sets of their game **(pure) strategies** -X of Alice, Y of Bob. Mappings $p_X : X \times Y \to \mathbf{R}$ and $p_Y : X \times Y \to \mathbf{R}$ will be called **payoff functions** of Alice and Bob. The quadruple (X, Y, p_X, p_Y) will be called a (mathematical) **game**.

A **mixed strategy** will be a probability distribution on pure strategies.

BASIC CONCEPTS of CLASSICAL GAME THEORY

We will consider **games with two players**, Alice and Bob. X and Y will be nonempty sets of their game **(pure) strategies** -X of Alice, Y of Bob. Mappings $p_X : X \times Y \rightarrow \mathbf{R}$ and $p_Y : X \times Y \rightarrow \mathbf{R}$ will be called **payoff functions** of Alice and Bob. The quadruple (X, Y, p_X, p_Y) will be called a (mathematical) **game**.

A **mixed strategy** will be a probability distribution on pure strategies.

An element $(x, y) \in X \times Y$ is said to be a **Nash equilibrium** of the game (X, Y, p_X, p_Y) iff $p_X(x', y) \leq p_X(x, y)$ for any $x' \in X$, and $p_Y(x, y') \leq p_Y(x, y)$ **for all** $y' \in Y$.

BASIC CONCEPTS of CLASSICAL GAME THEORY

We will consider **games with two players**, Alice and Bob. X and Y will be nonempty sets of their game **(pure) strategies** -X of Alice, Y of Bob. Mappings $p_X : X \times Y \rightarrow \mathbf{R}$ and $p_Y : X \times Y \rightarrow \mathbf{R}$ will be called **payoff functions** of Alice and Bob. The quadruple (X, Y, p_X, p_Y) will be called a (mathematical) **game**.

A **mixed strategy** will be a probability distribution on pure strategies.

An element $(x, y) \in X \times Y$ is said to be a **Nash equilibrium** of the game (X, Y, p_X, p_Y) iff $p_X(x', y) \leq p_X(x, y)$ for any $x' \in X$, and $p_Y(x, y') \leq p_Y(x, y)$ **for all** $y' \in Y$.

Informally, Nash equilibrium is such a pair of strategies that none of the players gains by changing his/her strategy.

BASIC CONCEPTS of CLASSICAL GAME THEORY

We will consider **games with two players**, Alice and Bob. X and Y will be nonempty sets of their game **(pure) strategies** -X of Alice, Y of Bob. Mappings $p_X : X \times Y \rightarrow \mathbf{R}$ and $p_Y : X \times Y \rightarrow \mathbf{R}$ will be called **payoff functions** of Alice and Bob. The quadruple (X, Y, p_X, p_Y) will be called a (mathematical) **game**.

A **mixed strategy** will be a probability distribution on pure strategies.

An element $(x, y) \in X \times Y$ is said to be a **Nash equilibrium** of the game (X, Y, p_X, p_Y) iff $p_X(x', y) \leq p_X(x, y)$ for any $x' \in X$, and $p_Y(x, y') \leq p_Y(x, y)$ **for all** $y' \in Y$.

Informally, Nash equilibrium is such a pair of strategies that none of the players gains by changing his/her strategy.

A game is called **zero-sum game** if $p_X(x, y) + p_Y(x, y) = 0$ for all $x \in X$ and $y \in Y$.

ONE of THE BASIC RESULTS

ONE of THE BASIC RESULTS

One of the basic result of the classical game theory is that not every two-players zero-sum game has a Nash equilibrium in the set of pure strategies, but there is always a Nash equilibrium if players follow mixed strategies.

POWER Of QUANTUM PHENOMENA

POWER Of QUANTUM PHENOMENA

It has been shown, for several zero-sum games, that if one of the players can use quantum tools and thereby **quantum strategies**, then he/she can increase his/her chance to win the game.

POWER Of QUANTUM PHENOMENA

It has been shown, for several zero-sum games, that if one of the players can use quantum tools and thereby **quantum strategies**, then he/she can increase his/her chance to win the game.

This way, from a fair game, in which both players have the same chance to win if only classical computation and communication tools are used, an unfair game can arise, or from an unfair game a fair one.

EXAMPLE - PENNY FLIP GAME

EXAMPLE - PENNY FLIP GAME

Alice and Bob play with a box and a penny as follows:

EXAMPLE - PENNY FLIP GAME

Alice and Bob play with a box and a penny as follows:

- Alice places a penny head up in a box.

EXAMPLE - PENNY FLIP GAME

Alice and Bob play with a box and a penny as follows:

- Alice places a penny head up in a box.
- Bob flips or does not flip the coin

EXAMPLE - PENNY FLIP GAME

Alice and Bob play with a box and a penny as follows:

- Alice places a penny head up in a box.
- Bob flips or does not flip the coin
- Alice flips or does not flip the coin

EXAMPLE - PENNY FLIP GAME

Alice and Bob play with a box and a penny as follows:

- Alice places a penny head up in a box.
- Bob flips or does not flip the coin
- Alice flips or does not flip the coin
- Bob flips or does not flip the coin

EXAMPLE - PENNY FLIP GAME

Alice and Bob play with a box and a penny as follows:

- Alice places a penny head up in a box.
- Bob flips or does not flip the coin
- Alice flips or does not flip the coin
- Bob flips or does not flip the coin

After the “game” is over, they open the box and Bob wins if the penny is head up.

EXAMPLE - PENNY FLIP GAME

Alice and Bob play with a box and a penny as follows:

- Alice places a penny head up in a box.
- Bob flips or does not flip the coin
- Alice flips or does not flip the coin
- Bob flips or does not flip the coin

After the "game" is over, they open the box and Bob wins if the penny is head up.

It is easy to check that using pure strategies chances to win are $\frac{1}{2}$ for each player and there is no (Nash) equilibrium in the case of pure classical strategies.

EXAMPLE - PENNY FLIP GAME

Alice and Bob play with a box and a penny as follows:

- Alice places a penny head up in a box.
- Bob flips or does not flip the coin
- Alice flips or does not flip the coin
- Bob flips or does not flip the coin

After the “game” is over, they open the box and Bob wins if the penny is head up.

It is easy to check that using pure strategies chances to win are $\frac{1}{2}$ for each player and there is no (Nash) equilibrium in the case of pure classical strategies.

However, there is equilibrium if Alice chooses its strategy with probability $\frac{1}{2}$ and Bob chooses each of the four possible strategies with probability $\frac{1}{4}$.

VERSION of PRISONERS' DILEMMA from 1992

VERSION of PRISONERS' DILEMMA from 1992

Two members of a gang are imprisoned, each in a separate cell, without possibility to communicate.

VERSION of PRISONERS' DILEMMA from 1992

Two members of a gang are imprisoned, each in a separate cell, without possibility to communicate. However, police has not enough evidence to convict them on the principal charge and therefore police intends to put both of them for one year to jail on a lesser charge.

VERSION of PRISONERS' DILEMMA from 1992

Two members of a gang are imprisoned, each in a separate cell, without possibility to communicate. However, police has not enough evidence to convict them on the principal charge and therefore police intends to put both of them for one year to jail on a lesser charge.

Simultaneously police offer both of them so called Faustian bargain.

VERSION of PRISONERS' DILEMMA from 1992

Two members of a gang are imprisoned, each in a separate cell, without possibility to communicate. However, police has not enough evidence to convict them on the principal charge and therefore police intends to put both of them for one year to jail on a lesser charge.

Simultaneously police offer both of them so called Faustian bargain. Each prisoner gets a chance either to betray the other one by testifying that he committed the crime,

VERSION of PRISONERS' DILEMMA from 1992

Two members of a gang are imprisoned, each in a separate cell, without possibility to communicate. However, police has not enough evidence to convict them on the principal charge and therefore police intends to put both of them for one year to jail on a lesser charge.

Simultaneously police offer both of them so called Faustian bargain. Each prisoner gets a chance either to betray the other one by testifying that he committed the crime, or to cooperate with the other one by remaining silent.

VERSION of PRISONERS' DILEMMA from 1992

Two members of a gang are imprisoned, each in a separate cell, without possibility to communicate. However, police has not enough evidence to convict them on the principal charge and therefore police intends to put both of them for one year to jail on a lesser charge.

Simultaneously police offer both of them so called Faustian bargain. Each prisoner gets a chance either to betray the other one by testifying that he committed the crime, or to cooperate with the other one by remaining silent. Here are payoffs they are offered:

VERSION of PRISONERS' DILEMMA from 1992

Two members of a gang are imprisoned, each in a separate cell, without possibility to communicate. However, police has not enough evidence to convict them on the principal charge and therefore police intends to put both of them for one year to jail on a lesser charge.

Simultaneously police offer both of them so called Faustian bargain. Each prisoner gets a chance either to betray the other one by testifying that he committed the crime, or to cooperate with the other one by remaining silent. Here are payoffs they are offered:

- If both betray, they will get into jail for 2 years.

VERSION of PRISONERS' DILEMMA from 1992

Two members of a gang are imprisoned, each in a separate cell, without possibility to communicate. However, police has not enough evidence to convict them on the principal charge and therefore police intends to put both of them for one year to jail on a lesser charge.

Simultaneously police offer both of them so called Faustian bargain. Each prisoner gets a chance either to betray the other one by testifying that he committed the crime, or to cooperate with the other one by remaining silent. Here are payoffs they are offered:

- If both betray, they will get into jail for 2 years.
- If one betrays and second decides to cooperate, then first will get free and second will go to jail for 3 years.

VERSION of PRISONERS' DILEMMA from 1992

Two members of a gang are imprisoned, each in a separate cell, without possibility to communicate. However, police has not enough evidence to convict them on the principal charge and therefore police intends to put both of them for one year to jail on a lesser charge.

Simultaneously police offer both of them so called Faustian bargain. Each prisoner gets a chance either to betray the other one by testifying that he committed the crime, or to cooperate with the other one by remaining silent. Here are payoffs they are offered:

- If both betray, they will get into jail for 2 years.
- If one betrays and second decides to cooperate, then first will get free and second will go to jail for 3 years.
- If both cooperate they will go to jail for 1 year.

VERSION of PRISONERS' DILEMMA from 1992

Two members of a gang are imprisoned, each in a separate cell, without possibility to communicate. However, police has not enough evidence to convict them on the principal charge and therefore police intends to put both of them for one year to jail on a lesser charge.

Simultaneously police offer both of them so called Faustian bargain. Each prisoner gets a chance either to betray the other one by testifying that he committed the crime, or to cooperate with the other one by remaining silent. Here are payoffs they are offered:

- If both betray, they will get into jail for 2 years.
- If one betrays and second decides to cooperate, then first will get free and second will go to jail for 3 years.
- If both cooperate they will go to jail for 1 year.

What is the best way for them to behave?

VERSION of PRISONERS' DILEMMA from 1992

Two members of a gang are imprisoned, each in a separate cell, without possibility to communicate. However, police has not enough evidence to convict them on the principal charge and therefore police intends to put both of them for one year to jail on a lesser charge.

Simultaneously police offer both of them so called Faustian bargain. Each prisoner gets a chance either to betray the other one by testifying that he committed the crime, or to cooperate with the other one by remaining silent. Here are payoffs they are offered:

- If both betray, they will get into jail for 2 years.
- If one betrays and second decides to cooperate, then first will get free and second will go to jail for 3 years.
- If both cooperate they will go to jail for 1 year.

What is the best way for them to behave? This game is a model for a variety of real-life situations involving cooperative behaviour.

VERSION of PRISONERS' DILEMMA from 1992

Two members of a gang are imprisoned, each in a separate cell, without possibility to communicate. However, police has not enough evidence to convict them on the principal charge and therefore police intends to put both of them for one year to jail on a lesser charge.

Simultaneously police offer both of them so called Faustian bargain. Each prisoner gets a chance either to betray the other one by testifying that he committed the crime, or to cooperate with the other one by remaining silent. Here are payoffs they are offered:

- If both betray, they will get into jail for 2 years.
- If one betrays and second decides to cooperate, then first will get free and second will go to jail for 3 years.
- If both cooperate they will go to jail for 1 year.

What is the best way for them to behave? This game is a model for a variety of real-life situations involving cooperative behaviour. Game was originally framed in 1950 by M. Flood and M. Dresher

PRISONERS' DILEMMA - I.

Two prisoners, Alice and Bob, can use, independently, any of the following two strategies: **to cooperate** or **to defect** (not to cooperate).

PRISONERS' DILEMMA - I.

Two prisoners, Alice and Bob, can use, independently, any of the following two strategies: **to cooperate** or **to defect** (not to cooperate).

The problem is that the payoff function (p_A, p_B), in millions, is a very special one (first (second) value is payoff of Alice (of Bob):

$\frac{\text{Alice}}{\text{Bob}}$	C_A	D_A
C_B	$(3,3)$	$(5,0)$
D_B	$(0,5)$	$(1,1)$

PRISONERS' DILEMMA - I.

Two prisoners, Alice and Bob, can use, independently, any of the following two strategies: **to cooperate** or **to defect** (not to cooperate).

The problem is that the payoff function (p_A, p_B), in millions, is a very special one (first (second) value is payoff of Alice (of Bob):

$\frac{\text{Alice}}{\text{Bob}}$	C_A	D_A
C_B	$(3,3)$	$(5,0)$
D_B	$(0,5)$	$(1,1)$

What is the best way for Alice and Bob to proceed in order to maximize their payoffs?

PRISONERS' DILEMMA - II.

A strategy s_A is called **dominant** for Alice if for any other strategy s'_A of Alice and s_B of Bob, it holds

$$P_A(s_A, s_B) \geq P_A(s'_A, s_B).$$

PRISONERS' DILEMMA - II.

A strategy s_A is called **dominant** for Alice if for any other strategy s'_A of Alice and s_B of Bob, it holds

$$P_A(s_A, s_B) \geq P_A(s'_A, s_B).$$

Clearly, defection is the dominant strategy of Alice (and also of Bob) in the case of Prisoners Dilemma game.

PRISONERS' DILEMMA - II.

A strategy s_A is called **dominant** for Alice if for any other strategy s'_A of Alice and s_B of Bob, it holds

$$P_A(s_A, s_B) \geq P_A(s'_A, s_B).$$

Clearly, defection is the dominant strategy of Alice (and also of Bob) in the case of Prisoners Dilemma game.

Prisoners Dilemma game has therefore
dominant-strategy equilibrium

PRISONERS' DILEMMA - II.

A strategy s_A is called **dominant** for Alice if for any other strategy s'_A of Alice and s_B of Bob, it holds

$$P_A(s_A, s_B) \geq P_A(s'_A, s_B).$$

Clearly, defection is the dominant strategy of Alice (and also of Bob) in the case of Prisoners Dilemma game.

Prisoners Dilemma game has therefore
dominant-strategy equilibrium

$\frac{\text{Alice}}{\text{Bob}}$	C_A	D_A
C_B	$(3,3)$	$(5,0)$
D_B	$(0,5)$	$(1,1)$

BATTLE of SEX GAME

BATTLE of SEX GAME

Alice and Bob have to decide, independently of each other, where to spent the evening.

BATTLE of SEX GAME

Alice and Bob have to decide, independently of each other, where to spent the evening.

Alice prefers to go to opera (O), Bob wants to watch TV (T) - tennis.

BATTLE of SEX GAME

Alice and Bob have to decide, independently of each other, where to spent the evening.

Alice prefers to go to opera (O), Bob wants to watch TV (T) - tennis.

However, at the same time both of them prefer to be together than to be apart.

BATTLE of SEX GAME

Alice and Bob have to decide, independently of each other, where to spent the evening.

Alice prefers to go to opera (O), Bob wants to watch TV (T) - tennis.

However, at the same time both of them prefer to be together than to be apart.

Pay-off function is given by the matrix (columns are for Alice) (columns are for Bob)

$$\begin{array}{ccc} & O & T \\ O & (\alpha, \beta) & (\gamma, \gamma) \\ T & (\gamma, \gamma) & (\beta, \alpha) \end{array}$$

where $\alpha > \beta > \gamma$.

BATTLE of SEX GAME

Alice and Bob have to decide, independently of each other, where to spent the evening.

Alice prefers to go to opera (O), Bob wants to watch TV (T) - tennis.

However, at the same time both of them prefer to be together than to be apart.

Pay-off function is given by the matrix (columns are for Alice) (columns are for Bob)

$$\begin{array}{ccc} & O & T \\ O & (\alpha,\beta) & (\gamma,\gamma) \\ T & (\gamma,\gamma) & (\beta,\alpha) \end{array}$$

where $\alpha > \beta > \gamma$.

What kind of strategy they should choose?

BATTLE of SEX GAME

Alice and Bob have to decide, independently of each other, where to spent the evening.

Alice prefers to go to opera (O), Bob wants to watch TV (T) - tennis.

However, at the same time both of them prefer to be together than to be apart.

Pay-off function is given by the matrix (columns are for Alice) (columns are for Bob)

$$\begin{array}{ccc} & O & T \\ O & (\alpha,\beta) & (\gamma,\gamma) \\ T & (\gamma,\gamma) & (\beta,\alpha) \end{array}$$

where $\alpha > \beta > \gamma$.

What kind of strategy they should choose?

The two Nash equilibria are (O, O) and (T, T), but players are faced with tactics dilemma, because these equilibria bring them different payoffs.

COIN GAME

COIN GAME

There are three coins: one fair, with both sides different, and two unfair, one with two heads and one with two tails.

COIN GAME

There are three coins: one fair, with both sides different, and two unfair, one with two heads and one with two tails.

The game proceeds as follows.

COIN GAME

There are three coins: one fair, with both sides different, and two unfair, one with two heads and one with two tails.

The game proceeds as follows.

- Alice puts coins into a black box and shakes the box.

COIN GAME

There are three coins: one fair, with both sides different, and two unfair, one with two heads and one with two tails.

The game proceeds as follows.

- Alice puts coins into a black box and shakes the box.
- Bob picks up one coin.

COIN GAME

There are three coins: one fair, with both sides different, and two unfair, one with two heads and one with two tails.

The game proceeds as follows.

- Alice puts coins into a black box and shakes the box.
- Bob picks up one coin.
- Alice wins if coin is unfair, otherwise Bob wins

Clearly, in the classical case, the probability that Alice wins is $\frac{2}{3}$.

FROM GAMES to LOWER BOUNDS for RANDOMIZED ALGORITHMS

FROM GAMES to LOWER BOUNDS for RANDOMIZED ALGORITHMS

Next goal is to present, using zero-sum games theory, a method how to prove lower bounds for the average running time of randomized algorithms.

FROM GAMES to LOWER BOUNDS for RANDOMIZED ALGORITHMS

Next goal is to present, using zero-sum games theory, a method how to prove lower bounds for the average running time of randomized algorithms.

This techniques can be applied to algorithms that terminate for all inputs and all random choices.

TWO–PERSON ZERO–SUM GAMES – EXAMPLE

TWO–PERSON ZERO–SUM GAMES – EXAMPLE

A two players zero–sum game is represented by an $n \times m$ payoff–matrix M with all rows and columns summing up to 0.

TWO–PERSON ZERO–SUM GAMES – EXAMPLE

A two players zero–sum game is represented by an $n \times m$ payoff–matrix M with all rows and columns summing up to 0.
Payoffs for n possible strategies of Alice are given in rows of M.

TWO–PERSON ZERO–SUM GAMES – EXAMPLE

A two players zero–sum game is represented by an $n \times m$ payoff–matrix M with all rows and columns summing up to 0.

Payoffs for n possible strategies of Alice are given in rows of M.

Payoffs for m possible strategies of Bob are given in columns of M.

TWO–PERSON ZERO–SUM GAMES – EXAMPLE

A two players zero–sum game is represented by an $n \times m$ payoff–matrix M with all rows and columns summing up to 0.

Payoffs for n possible strategies of Alice are given in rows of M.

Payoffs for m possible strategies of Bob are given in columns of M.

$$M_{i,j}$$

is the amount paid by Bob to Alice if Alice chooses strategy i and Bob's choice is strategy j.

TWO–PERSON ZERO–SUM GAMES – EXAMPLE

A two players zero–sum game is represented by an $n \times m$ payoff–matrix M with all rows and columns summing up to 0.

Payoffs for n possible strategies of Alice are given in rows of M.

Payoffs for m possible strategies of Bob are given in columns of M.

$$M_{i,j}$$

is the amount paid by Bob to Alice if Alice chooses strategy i and Bob's choice is strategy j.

The goal of Alice (Bob) is to maximize (minimize) her payoff.

TWO–PERSON ZERO–SUM GAMES – EXAMPLE

A two players zero–sum game is represented by an $n \times m$ payoff–matrix M with all rows and columns summing up to 0.
Payoffs for n possible strategies of Alice are given in rows of M.
Payoffs for m possible strategies of Bob are given in columns of M.

$$M_{i,j}$$

is the amount paid by Bob to Alice if Alice chooses strategy i and Bob's choice is strategy j.
The goal of Alice (Bob) is to maximize (minimize) her payoff.

Example - stone-scissors-paper game

PAYOFF–MATRIX

Bob

		Scissors	Paper	Stone
Alice	Scissors	0	1	-1
	Paper	-1	0	1
	Stone	1	-1	0

→ Table shows how much Bob has to pay to Alice

STRATEGIES for ZERO-INFORMATION and ZERO-SUM GAMES

STRATEGIES for ZERO-INFORMATION and ZERO-SUM GAMES

(Games with players having no information about their opponents' strategies.)

STRATEGIES for ZERO-INFORMATION and ZERO-SUM GAMES

(Games with players having no information about their opponents' strategies.)

Observe that if Alice chooses a strategy i, then she is guaranteed a payoff of $\min_j M_{ij}$ regardless of Bob's strategy.

STRATEGIES for ZERO-INFORMATION and ZERO-SUM GAMES

(Games with players having no information about their opponents' strategies.)

Observe that if Alice chooses a strategy i, then she is guaranteed a payoff of $\min_j M_{ij}$ regardless of Bob's strategy.

An optimal strategy O_A for Alice is such an i that maximises $\min_j M_{ij}$.

STRATEGIES for ZERO-INFORMATION and ZERO-SUM GAMES

(Games with players having no information about their opponents' strategies.)

Observe that if Alice chooses a strategy i, then she is guaranteed a payoff of $\min_j M_{ij}$ regardless of Bob's strategy.

An optimal strategy O_A for Alice is such an i that maximises $\min_j M_{ij}$.

$$O_A = \max_i \min_j M_{ij}$$

denotes therefore the lower bound on the value of the payoff Alice gains (from Bob) when she uses an optimal strategy.

STRATEGIES for ZERO-INFORMATION and ZERO-SUM GAMES

(Games with players having no information about their opponents' strategies.)

Observe that if Alice chooses a strategy i, then she is guaranteed a payoff of $\min_j M_{ij}$ regardless of Bob's strategy.

An optimal strategy O_A for Alice is such an i that maximises $\min_j M_{ij}$.

$$O_A = \max_i \min_j M_{ij}$$

denotes therefore the lower bound on the value of the payoff Alice gains (from Bob) when she uses an optimal strategy.

An optimal strategy O_B for Bob is such a j that minimizes $\max_i M_{ij}$.

STRATEGIES for ZERO-INFORMATION and ZERO-SUM GAMES

(Games with players having no information about their opponents' strategies.)

Observe that if Alice chooses a strategy i, then she is guaranteed a payoff of $\min_j M_{ij}$ regardless of Bob's strategy.

An optimal strategy O_A for Alice is such an i that maximises $\min_j M_{ij}$.

$$O_A = \max_i \min_j M_{ij}$$

denotes therefore the lower bound on the value of the payoff Alice gains (from Bob) when she uses an optimal strategy.

An optimal strategy O_B for Bob is such a j that minimizes $\max_i M_{ij}$. Bob's optimal strategy ensures therefore that his payoff is at least

$$O_B = \min_j \max_i M_{ij}$$

Theorem

$$O_A = \max_i \min_j M_{ij} \leq \min_j \max_i M_{ij} = O_B$$

Theorem

$$O_A = \max_i \min_j M_{ij} \leq \min_j \max_i M_{ij} = O_B$$

Often $O_A < O_B$. In our last (scissors-...) example, $-1 = O_A < O_B = +1$.

Theorem

$$O_A = \max_i \min_j M_{ij} \leq \min_j \max_i M_{ij} = O_B$$

Often $O_A < O_B$. In our last (scissors-...) example, $-1 = O_A < O_B = +1$.

If $O_B = O_A$ we say that the game has a solution – a specific choice of strategies that leads to this solution.

Theorem

$$O_A = \max_i \min_j M_{ij} \leq \min_j \max_i M_{ij} = O_B$$

Often $O_A < O_B$. In our last (scissors-...) example, $-1 = O_A < O_B = +1$.

If $O_B = O_A$ we say that the game has a solution – a specific choice of strategies that leads to this solution.

ϱ and γ are so called optional strategies for Alice and Bob if

$$O_A = O_B = M_{\varrho\gamma}$$

Example of the game which has a solution ($O_A = O_B = 0$)

$$\begin{matrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{matrix}$$

Example of the game which has a solution ($O_A = O_B = 0$)

0	1	2
-1	0	1
-2	-1	0

What happens if a game has no solution ?

Example of the game which has a solution ($O_A = O_B = 0$)

0	1	2
-1	0	1
-2	-1	0

What happens if a game has no solution ?

There is no clear–cut strategy for any player.

Example of the game which has a solution ($O_A = O_B = 0$)

0	1	2
-1	0	1
-2	-1	0

What happens if a game has no solution ?

There is no clear–cut strategy for any player.

Way out: to use randomized strategies.

Example of the game which has a solution ($O_A = O_B = 0$)

0	1	2
-1	0	1
-2	-1	0

What happens if a game has no solution ?

There is no clear–cut strategy for any player.

Way out: to use randomized strategies.

Alice chooses strategies according to a probability vector $p = (p_1, \ldots, p_n)$; p_i is probability that Alice chooses strategy $s_{A,i}$

Example of the game which has a solution ($O_A = O_B = 0$)

0	1	2
-1	0	1
-2	-1	0

What happens if a game has no solution ?

There is no clear–cut strategy for any player.

Way out: to use randomized strategies.

Alice chooses strategies according to a probability vector $p = (p_1, \ldots, p_n)$; p_i is probability that Alice chooses strategy $s_{A,i}$

Bob chooses strategies according to a probability vector $q = (q_1, \ldots, q_n)$; q_j is a probability that Bob chooses strategy $s_{B,j}$.

Example of the game which has a solution ($O_A = O_B = 0$)

$$\begin{array}{ccc} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{array}$$

What happens if a game has no solution ?
There is no clear–cut strategy for any player.
Way out: to use randomized strategies.

Alice chooses strategies according to a probability vector $p = (p_1, \ldots, p_n)$; p_i is probability that Alice chooses strategy $s_{A,i}$

Bob chooses strategies according to a probability vector $q = (q_1, \ldots, q_n)$; q_j is a probability that Bob chooses strategy $s_{B,j}$.

Payoff is now a random variable – **if p, q are taken as column vectors** then

$$E[\text{payoff}] = p^T Mq = \sum_{i=1}^{n} \sum_{j=1}^{m} p_i M_{ij} q_j$$

Let O_A (O_B) denote the best possible (optimal) lower (upper) bound on the expected payoff of Alice (Bob).

Let O_A (O_B) denote the best possible (optimal) lower (upper) bound on the expected payoff of Alice (Bob). Then it holds:

Let O_A (O_B) denote the best possible (optimal) lower (upper) bound on the expected payoff of Alice (Bob). Then it holds:

$$O_A = \max_p \min_q p^T Mq \quad O_B = \min_q \max_p p^T M$$

Theorem (von Neumann Minimax theorem) For any two–person zero–sum game specified by a payoff matrix M it holds

$$\max_p \min_q p^T Mq = \min_q \max_p p^T Mq$$

Theorem (von Neumann Minimax theorem) For any two–person zero–sum game specified by a payoff matrix M it holds

$$\max_{p} \min_{q} p^T Mq = \min_{q} \max_{p} p^T Mq$$

Observe that once p is fixed, $\max_p \min_q p^T Mq = \min_q \max_p p^T Mq$ is a linear function and is minimized by setting to 1 the q_j with the smallest coefficient in this linear function.

Theorem (von Neumann Minimax theorem) For any two–person zero–sum game specified by a payoff matrix M it holds

$$\max_p \min_q p^T Mq = \min_q \max_p p^T Mq$$

Observe that once p is fixed, $\max_p \min_q p^T Mq = \min_q \max_p p^T Mq$ is a linear function and is minimized by setting to 1 the q_j with the smallest coefficient in this linear function.

This has interesting/important implications:

Theorem (von Neumann Minimax theorem) For any two–person zero–sum game specified by a payoff matrix M it holds

$$\max_p \min_q p^T Mq = \min_q \max_p p^T Mq$$

Observe that once p is fixed, $\max_p \min_q p^T Mq = \min_q \max_p p^T Mq$ is a linear function and is minimized by setting to 1 the q_j with the smallest coefficient in this linear function.

This has interesting/important implications:

If Bob knows the distribution p used by Alice, then his optimal strategy is a pure strategy.

Theorem (von Neumann Minimax theorem) For any two–person zero–sum game specified by a payoff matrix M it holds

$$\max_p \min_q p^T Mq = \min_q \max_p p^T Mq$$

Observe that once p is fixed, $\max_p \min_q p^T Mq = \min_q \max_p p^T Mq$ is a linear function and is minimized by setting to 1 the q_j with the smallest coefficient in this linear function.

This has interesting/important implications:

If Bob knows the distribution p used by Alice, then his optimal strategy is a pure strategy.

A similar comment applies in the opposite direction.

Theorem (von Neumann Minimax theorem) For any two–person zero–sum game specified by a payoff matrix M it holds

$$\max_p \min_q p^T Mq = \min_q \max_p p^T Mq$$

Observe that once p is fixed, $\max_p \min_q p^T Mq = \min_q \max_p p^T Mq$ is a linear function and is minimized by setting to 1 the q_j with the smallest coefficient in this linear function.

This has interesting/important implications:

If Bob knows the distribution p used by Alice, then his optimal strategy is a pure strategy.

A similar comment applies in the opposite direction. This leads to a simplified version of the minimax theorem, where e_k denotes a unit vector with 1 at the k-th position and 0 elsewhere.

YAO'S TECHNIQUE 1/3

YAO'S TECHNIQUE 1/3

Yao's technique provides an application of the game-theoretic results to the establishment of lower bounds for randomized algorithms.

YAO'S TECHNIQUE 1/3

Yao's technique provides an application of the game-theoretic results to the establishment of lower bounds for randomized algorithms.
For a given algorithmic problem $\mathcal{P}$ let us consider the following payoff matrix.

deterministic algorithms

$$\begin{array}{cc|c|}
 & & \mathcal{A}_1 \quad \mathcal{A}_2 \quad \mathcal{A}_3 \\
\text{I} & c_1 & \\
\text{N} & c_2 & \text{entries} \\
\text{P} & c_3 & = \\
\text{U} & c_4 & \text{resources} \\
\text{T} & & \text{(i.e. used computation time)} \\
\text{S} & &
\end{array}$$

Bob – a designer
choosing good algorithms

Alice – an adversary
choosing bad inputs

YAO'S TECHNIQUE 1/3

Yao's technique provides an application of the game-theoretic results to the establishment of lower bounds for randomized algorithms.
For a given algorithmic problem $\mathcal{P}$ let us consider the following payoff matrix.

deterministic algorithms

$\mathcal{A}_1 \quad \mathcal{A}_2 \quad \mathcal{A}_3$

I c_1
N c_2
P c_3
U c_4
T
S

entries
=
resources
(i.e. used computation time)

Bob – a designer
choosing good algorithms

Alice – an adversary
choosing bad inputs

Pure strategy for Bob corresponds to the choice of a deterministic algorithm.

YAO'S TECHNIQUE 1/3

Yao's technique provides an application of the game-theoretic results to the establishment of lower bounds for randomized algorithms.
For a given algorithmic problem $\mathcal{P}$ let us consider the following payoff matrix.

deterministic algorithms

$\mathcal{A}_1$ $\mathcal{A}_2$ $\mathcal{A}_3$

INPUTS

c_1
c_2
c_3
c_4

entries
=
resources
(i.e. used computation time)

Bob – a designer
choosing good algorithms

Alice – an adversary
choosing bad inputs

Pure strategy for Bob corresponds to the choice of a deterministic algorithm.
Optimal pure strategy for Bob corresponds to a choice of an optimal deterministic algorithm.

YAO'S TECHNIQUE 2/3

Let V_B be the worst-case running time of any deterministic algorithm of Bob

YAO'S TECHNIQUE 2/3

Let V_B be the worst-case running time of any deterministic algorithm of Bob

Problem: How to interpret mixed strategies ?

YAO'S TECHNIQUE 2/3

Let V_B be the worst-case running time of any deterministic algorithm of Bob

Problem: How to interpret mixed strategies ?

A *mixed strategy* for Bob is a probability distribution over (always correct) deterministic algorithms—so it is a Las Vegas randomized algorithm.

YAO'S TECHNIQUE 2/3

Let V_B be the worst-case running time of any deterministic algorithm of Bob

Problem: How to interpret mixed strategies ?

A *mixed strategy* for Bob is a probability distribution over (always correct) deterministic algorithms—so it is a Las Vegas randomized algorithm.

An optimal mixed strategy for Bob is an optimal Las Vegas algorithm.
Distributional complexity of a problem is an expected running time of the best deterministic algorithm for the worst distribution on the inputs.

YAO'S TECHNIQUE 2/3

Let V_B be the worst-case running time of any deterministic algorithm of Bob

Problem: How to interpret mixed strategies ?

A *mixed strategy* for Bob is a probability distribution over (always correct) deterministic algorithms—so it is a Las Vegas randomized algorithm.

An optimal mixed strategy for Bob is an optimal Las Vegas algorithm.
Distributional complexity of a problem is an expected running time of the best deterministic algorithm for the worst distribution on the inputs.

Loomis theorem implies that distributional complexity equals to the least possible time achievable by any randomized algorithm

Reformulation of von Neumann+Loomis' theorem in the language of algorithms

Reformulation of von Neumann+Loomis' theorem in the language of algorithms

Corollary Let Π be a problem with a finite set I of input instances and $\mathcal{A}$ be a finite set od deterministic algorithms for Π.

Reformulation of von Neumann+Loomis' theorem in the language of algorithms

Corollary Let Π be a problem with a finite set I of input instances and $\mathcal{A}$ be a finite set od deterministic algorithms for Π. For any input $i \in I$ and any algorithm $A \in \mathcal{A}$, let $T(i, A)$ denote computation time of A on input i.

Reformulation of von Neumann+Loomis' theorem in the language of algorithms

Corollary Let Π be a problem with a finite set I of input instances and $\mathcal{A}$ be a finite set od deterministic algorithms for Π. For any input $i \in I$ and any algorithm $A \in \mathcal{A}$, let $T(i, A)$ denote computation time of A on input i. For probability distributions p over I and q over $\mathcal{A}$, let i_p denote random input chosen according to p and A_q a random algorithm chosen according to q.

Reformulation of von Neumann+Loomis' theorem in the language of algorithms

Corollary Let Π be a problem with a finite set I of input instances and $\mathcal{A}$ be a finite set od deterministic algorithms for Π. For any input $i \in I$ and any algorithm $A \in \mathcal{A}$, let $T(i, A)$ denote computation time of A on input i. For probability distributions p over I and q over $\mathcal{A}$, let i_p denote random input chosen according to p and A_q a random algorithm chosen according to q. Then

$$\max_p \min_q E\left[T(i_p, A_q)\right] = \min_q \max_p E\left[T(i_p, A_q)\right]$$

$$\max_p \min_{A \in \mathcal{A}} E\left[T(i_p, A)\right] = \min_q \max_{i \in I} E\left[T(i, A_q)\right]$$

YAO'S TECHNIQUE 3/3

YAO'S TECHNIQUE 3/3

Consequence:

YAO'S TECHNIQUE 3/3

Consequence:

Theorem(Yao's Minimax Principle) For all distributions p over I and q over $\mathcal{A}$.

$$\min_{A \in \mathcal{A}} \mathbf{E}[T(i_p, A)] \leq \max_{i \in I} \mathbf{E}[T(i, A_q)]$$

YAO'S TECHNIQUE 3/3

Consequence:

Theorem(Yao's Minimax Principle) For all distributions p over I and q over $\mathcal{A}$.

$$\min_{A \in \mathcal{A}} \mathbf{E}[T(i_p, A)] \leq \max_{i \in I} \mathbf{E}[T(i, A_q)]$$

Interpretation: Expected running time of the optimal deterministic algorithm for an arbitrarily chosen input distribution p for a problem Π is a lower bound on the expected running time of the optimal (Las Vegas) randomized algorithm for Π.

YAO'S TECHNIQUE 3/3

Consequence:

Theorem(Yao's Minimax Principle) For all distributions p over I and q over $\mathcal{A}$.

$$\min_{A\in\mathcal{A}} \mathbf{E}[T(i_p, A)] \leq \max_{i\in I} \mathbf{E}[T(i, A_q)]$$

Interpretation: Expected running time of the optimal deterministic algorithm for an arbitrarily chosen input distribution p for a problem Π is a lower bound on the expected running time of the optimal (Las Vegas) randomized algorithm for Π.

In other words, to determine a lower bound on the performance of all randomized algorithms for a problem P, derive instead a lower bound for any deterministic algorithm for P when its inputs are drawn from a specific probability distribution (of your choice).

IMPLICATIONS OF YAO'S MINIMAX PRINCIPLE

IMPLICATIONS OF YAO'S MINIMAX PRINCIPLE

Interpretation again Expected running time of the optimal deterministic algorithm for an arbitrarily chosen input distribution p for a problem Π is a lower bound on the expected running time of the optimal (Las Vegas) randomized algorithm for Π.

IMPLICATIONS OF YAO'S MINIMAX PRINCIPLE

Interpretation again Expected running time of the optimal deterministic algorithm for an arbitrarily chosen input distribution p for a problem Π is a lower bound on the expected running time of the optimal (Las Vegas) randomized algorithm for Π.

Consequence:

In order to prove a lower bound on the randomized complexity of an algorithmic problem, it suffices to choose *any* probability distribution p on the input and prove a lower bound on the expected running time of deterministic algorithms for that distribution.

IMPLICATIONS OF YAO'S MINIMAX PRINCIPLE

Interpretation again Expected running time of the optimal deterministic algorithm for an arbitrarily chosen input distribution p for a problem Π is a lower bound on the expected running time of the optimal (Las Vegas) randomized algorithm for Π.

Consequence:

In order to prove a lower bound on the randomized complexity of an algorithmic problem, it suffices to choose *any* probability distribution p on the input and prove a lower bound on the expected running time of deterministic algorithms for that distribution.

The power of this technique lies in

1. the flexibility at the choice of p

IMPLICATIONS OF YAO'S MINIMAX PRINCIPLE

Interpretation again Expected running time of the optimal deterministic algorithm for an arbitrarily chosen input distribution p for a problem Π is a lower bound on the expected running time of the optimal (Las Vegas) randomized algorithm for Π.

Consequence:

In order to prove a lower bound on the randomized complexity of an algorithmic problem, it suffices to choose *any* probability distribution p on the input and prove a lower bound on the expected running time of deterministic algorithms for that distribution.

The power of this technique lies in

1. the flexibility at the choice of p
2. the reduction of the task to determine lower bounds for randomized algorithms to the task to determine lower bounds for deterministic algorithms.

(It is important to remember that we can expect that the deterministic algorithm "knows" the chosen distribution p.)

The above discussion holds for Las Vegas algorithms only!

THE CASE OF MONTE CARLO ALGORITHMS

Let us consider Monte Carlo algorithms with error probability $0 < \varepsilon < \frac{1}{2}$.

Let us define the **distributional complexity with error ε**, notation

$$\min_{A \in \mathcal{A}} E\left[T_{\varepsilon}(I_p, A)\right],$$

to be the minimum expected time of any deterministic algorithm that errs with probability at most ε under the input I_p with distribution p.

Let us denote by

$$\max_{i \in \mathcal{I}} E\left[(T_{\varepsilon}(i, A_q)\right]$$

the expected time (under the worst input) of any randomized algorithm A_q that errs with probability at most ε.

Theorem For all distributions p over inputs and q over Algorithms, and any $\varepsilon \in [0, 1/2]$, it holds

$$\frac{1}{2}(\min_{A \in \mathcal{A}} \mathbf{E}\left[T_{2\varepsilon}(i_p, A)\right]) \leq \max_{i \in \mathcal{I}} \mathbf{E}\left[T_{\varepsilon}(i, A_q)\right]$$

GAMES TREES REVISITED

GAMES TREES REVISITED

A randomized algorithm for a game-tree T evaluations can be viewed as a probability distribution over deterministic algorithms for T, because the length of computation and the number of choices at each step are finite.

GAMES TREES REVISITED

A randomized algorithm for a game-tree T evaluations can be viewed as a probability distribution over deterministic algorithms for T, because the length of computation and the number of choices at each step are finite.

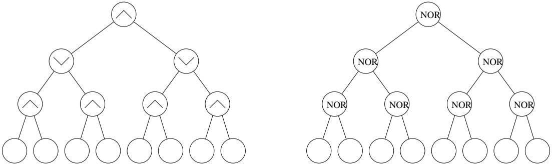
Instead of AND–OR trees of depth $2k$ we can consider NOR–trees of depth $2k$.

GAMES TREES REVISITED

A randomized algorithm for a game-tree T evaluations can be viewed as a probability distribution over deterministic algorithms for T, because the length of computation and the number of choices at each step are finite.

Instead of AND–OR trees of depth $2k$ **we can consider NOR–trees of depth** $2k$. Indeed, it holds:

$$(a \vee b) \wedge (c \vee d) \equiv (a \text{ NOR } b)\text{NOR}(c \text{ NOR } d)$$



Note: It's important to distinguish between:

Note: It's important to distinguish between:

- the expected running time of the randomized algorithm with a fixed input (where probability is considered over all random choices made by the algorithm)

Note: It's important to distinguish between:

- the expected running time of the randomized algorithm with a fixed input (where probability is considered over all random choices made by the algorithm)
- and

Note: It's important to distinguish between:

- the expected running time of the randomized algorithm with a fixed input (where probability is considered over all random choices made by the algorithm)
- and
- the expected running time of the deterministic algorithm when proving the lower bound (the average time is taken over all random input instances).

LOWER BOUND FOR GAME TREE EVALUATION - I

LOWER BOUND FOR GAME TREE EVALUATION - I

Assume now that each leaf of a NOR-tree is set up to have value 1 with probability $p = \frac{3-\sqrt{5}}{2}$

LOWER BOUND FOR GAME TREE EVALUATION - I

Assume now that each leaf of a NOR-tree is set up to have value 1 with probability $p = \frac{3-\sqrt{5}}{2}$ (observe that $(1-p)^2 = p$ for such a p).

LOWER BOUND FOR GAME TREE EVALUATION - I

Assume now that each leaf of a NOR-tree is set up to have value 1 with probability $p = \frac{3-\sqrt{5}}{2}$ (observe that $(1-p)^2 = p$ for such a p).

Observe that if inputs of a NOR-gate have value 1 with probability p then its output value is also 1 with probability $(1-p)(1-p) = p$.

LOWER BOUND FOR GAME TREE EVALUATION - I

Assume now that each leaf of a NOR-tree is set up to have value 1 with probability $p = \frac{3-\sqrt{5}}{2}$ (observe that $(1-p)^2 = p$ for such a p).

Observe that if inputs of a NOR-gate have value 1 with probability p then its output value is also 1 with probability $(1-p)(1-p) = p$.

Consider now only *depth–first pruning algorithms* for tree evaluation.
(They are such depth–first algorithms that make use of the knowledge that subtrees that provide no additional useful information can be "pruned away".)

LOWER BOUND FOR GAME TREE EVALUATION - I

Assume now that each leaf of a NOR-tree is set up to have value 1 with probability $p = \frac{3-\sqrt{5}}{2}$ (observe that $(1-p)^2 = p$ for such a p).

Observe that if inputs of a NOR-gate have value 1 with probability p then its output value is also 1 with probability $(1-p)(1-p) = p$.

Consider now only *depth–first pruning algorithms* for tree evaluation.
(They are such depth–first algorithms that make use of the knowledge that subtrees that provide no additional useful information can be "pruned away".)

Of importance for the overall analysis is the following technical lemma:

LOWER BOUND FOR GAME TREE EVALUATION - I

Assume now that each leaf of a NOR-tree is set up to have value 1 with probability $p = \frac{3-\sqrt{5}}{2}$ (observe that $(1-p)^2 = p$ for such a p).

Observe that if inputs of a NOR-gate have value 1 with probability p then its output value is also 1 with probability $(1-p)(1-p) = p$.

Consider now only *depth–first pruning algorithms* for tree evaluation.
(They are such depth–first algorithms that make use of the knowledge that subtrees that provide no additional useful information can be "pruned away".)

Of importance for the overall analysis is the following technical lemma:

Lemma Let T be a NOR–tree each leaf of which is set to 1 with a fixed probability. Let $W(T)$ denote the minimum, over all deterministic algorithms, of the expected number of steps to evaluate T. Then there is a depth–first pruning algorithm whose expected number of steps to evaluate T is $W(T)$.

LOWER BOUND FOR GAME TREE EVALUATION - I

Assume now that each leaf of a NOR-tree is set up to have value 1 with probability $p = \frac{3-\sqrt{5}}{2}$ (observe that $(1-p)^2 = p$ for such a p).

Observe that if inputs of a NOR-gate have value 1 with probability p then its output value is also 1 with probability $(1-p)(1-p) = p$.

Consider now only *depth–first pruning algorithms* for tree evaluation.
(They are such depth–first algorithms that make use of the knowledge that subtrees that provide no additional useful information can be "pruned away".)

Of importance for the overall analysis is the following technical lemma:

Lemma Let T be a NOR–tree each leaf of which is set to 1 with a fixed probability. Let $W(T)$ denote the minimum, over all deterministic algorithms, of the expected number of steps to evaluate T. Then there is a depth–first pruning algorithm whose expected number of steps to evaluate T is $W(T)$.

The last lemma tells us that for the purposes of our lower bound, we may restrict our attention to the depth–first pruning algorithms.

LOWER BOUND FOR GAME TREE EVALUATION - II

For a depth–first pruning algorithm evaluating a NOR–tree, let $W(h)$ be the expected number of leaves the algorithm inspects in determining the value of a node at distance h from the leaves.

LOWER BOUND FOR GAME TREE EVALUATION - II

For a depth–first pruning algorithm evaluating a NOR–tree, let $W(h)$ be the expected number of leaves the algorithm inspects in determining the value of a node at distance h from the leaves.

It holds

$$W(h) = pW(h-1) + (1-p)2W(h-1) = (2-p)W(h-1)$$

because with the probability $1-p$ the first subtree produces 0 and therefore also the second tree has to be evaluated.

LOWER BOUND FOR GAME TREE EVALUATION - II

For a depth–first pruning algorithm evaluating a NOR–tree, let $W(h)$ be the expected number of leaves the algorithm inspects in determining the value of a node at distance h from the leaves.

It holds

$$W(h) = pW(h-1) + (1-p)2W(h-1) = (2-p)W(h-1)$$

because with the probability $1-p$ the first subtree produces 0 and therefore also the second tree has to be evaluated. If $h = \lg_2 n$, then the above recursion has a solution

$$W(h) \geq n^{0.694}.$$

LOWER BOUND FOR GAME TREE EVALUATION - II

For a depth–first pruning algorithm evaluating a NOR–tree, let $W(h)$ be the expected number of leaves the algorithm inspects in determining the value of a node at distance h from the leaves.

It holds

$$W(h) = pW(h-1) + (1-p)2W(h-1) = (2-p)W(h-1)$$

because with the probability $1-p$ the first subtree produces 0 and therefore also the second tree has to be evaluated. If $h = \lg_2 n$, then the above recursion has a solution

$$W(h) \geq n^{0.694}.$$

This implies:

LOWER BOUND FOR GAME TREE EVALUATION - II

For a depth–first pruning algorithm evaluating a NOR–tree, let $W(h)$ be the expected number of leaves the algorithm inspects in determining the value of a node at distance h from the leaves.

It holds

$$W(h) = pW(h-1) + (1-p)2W(h-1) = (2-p)W(h-1)$$

because with the probability $1-p$ the first subtree produces 0 and therefore also the second tree has to be evaluated. If $h = \lg_2 n$, then the above recursion has a solution

$$W(h) \geq n^{0.694}.$$

This implies:

Theorem The expected running time of any randomized algorithm that always evaluates an instance of T_k correctly is at least $n^{0.694}$, where $n = 2^{2k}$ is the number of leaves.

The upper bound for randomized game tree evaluation algorithms already shown, at the beginning of this chapter was $n^{0.79}$, what is more than the lower bound n^{694} just shown.

The upper bound for randomized game tree evaluation algorithms already shown, at the beginning of this chapter was $n^{0.79}$, what is more than the lower bound n^{694} just shown.

It was therefore natural to ask what does the previous theorem really says?

The upper bound for randomized game tree evaluation algorithms already shown, at the beginning of this chapter was $n^{0.79}$, what is more than the lower bound n^{694} just shown.

It was therefore natural to ask what does the previous theorem really says?

For example, is our lower bound technique weak? ?

The upper bound for randomized game tree evaluation algorithms already shown, at the beginning of this chapter was $n^{0.79}$, what is more than the lower bound n^{694} just shown.

It was therefore natural to ask what does the previous theorem really says?

For example, is our lower bound technique weak? ?

No, the above result just says that in order to get a better lower bound another probability distribution on inputs may be needed.

RECENT RESULTS

RECENT RESULTS

Two recent results put more light on the Game tree evaluation problem.

RECENT RESULTS

Two recent results put more light on the Game tree evaluation problem.

- It has been shown that for our game tree evaluation problem the upper bound presented at the beginning is the best possible and therefore that $\theta(n^{0.79})$ is indeed the classical (query) complexity of the problem.

RECENT RESULTS

Two recent results put more light on the Game tree evaluation problem.

- It has been shown that for our game tree evaluation problem the upper bound presented at the beginning is the best possible and therefore that $\theta(n^{0.79})$ is indeed the classical (query) complexity of the problem.
- It has also been shown, by Farhi et al. (2009), that the upper bound for the case quantum computation tools can be used is $O(n^{0.5})$.

APPENDIX

APPENDIX

The concept of the number of wisdom introduced in the following and related results helped to show that randomness is deeply rooted even in arithmetic.

APPENDIX

The concept of the number of wisdom introduced in the following and related results helped to show that randomness is deeply rooted even in arithmetic.

In order to define numbers of wisdom the concept of self-delimiting programs is needed.

APPENDIX

The concept of the number of wisdom introduced in the following and related results helped to show that randomness is deeply rooted even in arithmetic.

In order to define numbers of wisdom the concept of self-delimiting programs is needed.

A program represented by a binary word p, is self-delimiting for a computer C, if for any input pw the computer C can recognize where p ends after reading p only..

Another way to see self-delimiting programs is to consider only such programming languages L that no program in L is a prefix of another program in L.

Ω - numbers of wisdom

For a universal computer C with only self-delimiting programs, the number of wisdom Ω_C is the probability that randomly constructed program for C halts.

Ω - numbers of wisdom

For a universal computer C with only self-delimiting programs, the number of wisdom Ω_C is the probability that randomly constructed program for C halts. More formally

Ω - numbers of wisdom

For a universal computer C with only self-delimiting programs, the number of wisdom Ω_C is the probability that randomly constructed program for C halts. More formally

$$\Omega_C = \sum_{p \text{ halts}} 2^{-|p|}$$

where p are (self-delimiting) halting programs for C.

Ω - numbers of wisdom

For a universal computer C with only self-delimiting programs, the number of wisdom Ω_C is the probability that randomly constructed program for C halts. More formally

$$\Omega_C = \sum_{p \text{ halts}} 2^{-|p|}$$

where p are (self-delimiting) halting programs for C.

Ω_C is therefore the probability that a self-delimiting computer program for C generated at random, by choosing each of its bits using an independent toss of a fair coin, will eventually halt.

Properties of numbers of wisdom

Properties of numbers of wisdom

- $0 \leq \Omega_C \leq 1$

Properties of numbers of wisdom

- $0 \leq \Omega_C \leq 1$
- Ω_C is an uncomputable and random real number.

Properties of numbers of wisdom

- $0 \leq \Omega_C \leq 1$
- Ω_C is an uncomputable and random real number.
- At least n-bits long theory is needed to determine n bits of Ω_C.

Properties of numbers of wisdom

- $0 \leq \Omega_C \leq 1$
- Ω_C is an uncomputable and random real number.
- At least n-bits long theory is needed to determine n bits of Ω_C.
- At least n bits long program is needed to determine n bits of Ω_C
- Bits of Ω can be seen as mathematical facts that are true for no reason.

- Greg Chaitin, who introduced numbers of wisdom, designed a specific universal computer C and a two hundred pages long Diophantine equation E, with 17,000 variables and with one parameter k, such that for a given k the equation E has a finite (infinite) number of solutions if and only if the k-th bit of Ω_C is 0 (is 1).

- Greg Chaitin, who introduced numbers of wisdom, designed a specific universal computer C and a two hundred pages long Diophantine equation E, with 17,000 variables and with one parameter k, such that for a given k the equation E has a finite (infinite) number of solutions if and only if the k-th bit of Ω_C is 0 (is 1).{ As a consequence, we have that randomness, unpredictability and uncertainty occur even in the theory of Diophantine equations of elementary arithmetic.}

- Greg Chaitin, who introduced numbers of wisdom, designed a specific universal computer C and a two hundred pages long Diophantine equation E, with 17,000 variables and with one parameter k, such that for a given k the equation E has a finite (infinite) number of solutions if and only if the k-th bit of Ω_C is 0 (is 1).{ As a consequence, we have that randomness, unpredictability and uncertainty occur even in the theory of Diophantine equations of elementary arithmetic.}
- Knowing the value of Ω_C with n bits of precision allows to decide which programs for C with at most n bits halt.