

Information and Coding Theory (CO349)

Coding and Decoding

Herbert Wiklicky
herbert@doc.ic.ac.uk

Department of Computing
Imperial College London

Autumn 2018

Slide 1 of 40

Information and Codes

What will be covered in this course?

- Data and Information Representation
- Data and Information Transmission

We will look at these problems from a quantitative, statistical point of view as well as from an algebraic point of view.

In German, Dutch, Italian, etc: CS is Informatik – Informatica
Information Science rather than **Computer Science**.

Slide 2 of 40

Practicalities

Two Lecturers

Herbert Wiklicky Slides on `~herbert/teaching.html`

`h.wiklicky@imperial.ac.uk`

Teaching $3\frac{1}{2}$ weeks until 29 October

Open-book coursework test **24 October** (or 29?)

Mahdi Cheraghchi Slides on Piazza

`cheraghchi@gmail.com`

Teaching $3\frac{1}{2}$ weeks from 31 October

Open-book coursework test **21 November**

João Lourenco Ribeiro GTA, Piazza, etc.

Exam: Week 11, **13 December 2018**, 2 hours (3 out of 4).

Different classes, different background, different applications.

Slide 3 of 40

Information and Coding (Theory)

Part I: Information Theory

- Simple Codes and Coding
- Revision: Probabilities
- Information Representation
- Information Transmission
- [Information Manifestation]

Part II: Coding Theory

- Revision: Linear Algebra
- Linear Codes
- Algebraic Error-Correcting Codes
- [Information in Cryptography]

Slide 4 of 40

Text Books

- Norman L. Biggs: *Codes - An Introduction to Information, Communication and Cryptography*, Springer 2008.
- Ron Roth: *Introduction to Coding Theory*, Cambridge University Press, 2006.
- David Applebaum: *Probability and Information*, Cambridge University Press, 1996/2008.
- Robert McEliece: *The Theory of Information and Coding*, Cambridge University Press, 2002.
- T.M. Cover, J.A. Thomas: *Elements of Information Theory*, Wiley-Interscience, 2006.
- Robert B. Ash: *Information Theory*, Wiley 1965, reprint: Dover 1980.

Slide 5 of 40

Information and Security

Side-Channel Attacks (Kocher, 1996)

The problem appears in attacks against public encryption algorithms like RSA. In (optimised) versions of de/encoding (using modular exponentiation) properties of the secret key determine the execution time.

How much **information** about the secret key is revealed?

Differential Privacy (Dwork, 2006)

In large (statistical) databases an attacker can try to reveal information about individuals (de-anonymise), e.g. there are only 3 under-25 with hair loss registered, and there are two people getting hair-loss treatment in SW7. Kim is on the database, 21 and lives at 170 Queen's Gate.

How much **information** about individuals is revealed?

Slide 6 of 40

Information – A Measure of Surprise

The **Entropy** of a probability distribution \mathbf{p} is:

$$H(\mathbf{p}) = - \sum \mathbf{p}(x) \log_2(\mathbf{p}(x)).$$

Example (Cover&Thomas)

Consider a Cheltenham Horse Race with 8 horses with winning chances $(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64})$, then the **entropy** is 2.

Label horses 0, 10, 110, 1110, 111100, 111101, 111110, 111111 then we need only only 2 bits **in average** to report winner.

For equally strong horses $(\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8})$ we have an entropy of 3, and the minimal message length is also 3 bits.

Slide 7 of 40

Surprise vs Prejudice

A father and son are on a fishing trip in the mountains of Wales. On the way back home their car has a serious accident.

The father is immediately killed and declared dead on the site of the accident. However, the son is severely injured and driven by ambulance to the next hospital.

When the son is brought into the operating theatre the surgeon exclaims "I can't do this, he is my son."

Scientific American, 1980s

Slide 8 of 40

Everything You Always Wanted to Know About Logarithms But Were Afraid to Ask

Defined (implicitly) via the equation:

$$x = b^{\log_b(x)}$$

Common logarithms: $b = 2$, $b = 10$, and $b = 2.71828\dots = e$

$$\log_b(xy) = \log_b(x) + \log_b(y)$$

$$\log_b(x^n) = n \log_b(x), \dots \quad \log_b(x) = \frac{\log_a(x)}{\log_a(b)}$$

Example

$\log_2(64) = 6$ as $2^6 = 64$; $\log_2(\frac{1}{64}) = -6$ as $\frac{1}{64} = 2^{-6}$ ($2^{-1} = \frac{1}{2}$).

Slide 9 of 40

Messages and Strings

Definition

An **alphabet** is a finite set S ; its members $s \in S$ are referred to as **symbols**.

Definition

A **message** m in the alphabet S is a finite sequence of elements in S :

$$m = s_1 s_2 \cdots s_n \quad \text{with } s_i \in S, 1 \leq i \leq n.$$

A message is often also referred to as a **string** or **word** in S . The number $n \in \mathbb{N}$ is the **length** of m .

Notation for **length** $|m| = n$; **prefix** $m|_k = s_1 s_2 \cdots s_k$ (with $k \leq n$).

Slide 10 of 40

Notation

Consider also string of length zero, often denoted ε with $|\varepsilon| = 0$.

By S^0 we denote the set of words containing only ε ; S^n denotes the set of all strings of length n ; and

$$S^* = S^0 \cup S^1 \cup S^2 \cup \dots$$

Example

With the alphabet $S = \mathbb{B} = \{0, 1\}$ we can produce all binary messages of a certain length n . Concretely, for example:

$$\mathbb{B}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$$

Slide 11 of 40

Coding

Definition

Let S and T be two alphabets. A **code** c is an *injective* function

$$c : S \rightarrow T^*$$

For each symbol $s \in S$ the string $c(s) \in T^*$ is called the **codeword** for s . The set of all codewords

$$C = \{c(s) \mid s \in S\}$$

is also referred to as the **code**.

For $|T| = 2$, e.g. $T = \mathbb{B}$, we have a **binary**, for $|T| = 3$ a **ternary**, and for $|T| = b$ a **b-ary** code.

Slide 12 of 40

Example: Morse Code

Example

Development from 1836 by Samuel F. B. Morse (1791–1872).
This is a code $c : \{A, \dots, Z\} \rightarrow \{\bullet, -, \odot\}^*$ (with variations):

A	•—	H	••••	O	---	V	•••—
B	—•••	I	••	P	•---•	W	•---
C	—•—•	J	•----	Q	--•—	X	—••—
D	—••	K	—•—	R	•—•	Y	—•---
E	•	L	•—••	S	•••	Z	---••
F	••—•	M	--	T	—		
G	--•	N	—•	U	••—		⊙

Most famous (after April 1912) perhaps:

$SOS \mapsto \bullet\bullet\bullet\odot\text{---}\odot\bullet\bullet\bullet\odot \equiv \bullet\bullet\bullet\text{---}\bullet\bullet\bullet$

Slide 13 of 40

Decoding

Definition

A code $c : S \rightarrow T^*$ is extended to S^* as follows: Given a string $s_1 s_2 \cdots s_n$ in S^* we define

$$c(s_1 s_2 \cdots s_n) = c(s_1) c(s_2) \cdots c(s_n)$$

Note: This way we overload the function c (polymorphism).

Definition

A code $c : S \rightarrow T^*$ is **uniquely decodable** (short **UD**) if the extended code function $c : S^* \rightarrow T^*$ is *injective*.

This means that every string in T^* corresponds to at most one message in S^* .

Slide 14 of 40

Example

Example

Let $S = \{x, y, z\}$ and take a code $c : S \rightarrow \mathbb{B}^*$ defined as

$$x \mapsto 0 \quad y \mapsto 10 \quad z \mapsto 11$$

Take the **original stream** $yzxxzyxzyy \dots$ in S^* and encode it.

Is it always possible to decode the **encoded stream** in $T^* = \mathbb{B}^*$?

Example

Same code as above. Consider **encoded stream**

1000111000...

What is the original stream? Is it uniquely determined?

Slide 15 of 40

The Decoding Problem

Given a sequence in T^* can we (sequentially) decode it?

Example

Same code with $C = \{0, 10, 11\}$. Consider 100111000...

Take prefix of length 1, first **1**: it is **not** in the **codebook** C .

Take prefix of length 2, i.e. **10**: this matches with y .

Consider remainder stream 0111000...

Take prefix of length 1, i.e. **0**: this matches with x .

Consider remainder stream 111000...

Take prefix of length 1, i.e. **1**: is **not** in the **codebook** C .

Take prefix of length 2, i.e. **11**: this matches with z .

Example

Without the 'pause' \odot in the Morse code we could not decode it, e.g. IEEE as $\bullet \bullet \bullet \bullet \bullet$ vs $\bullet \bullet \odot \bullet \odot \bullet \odot \bullet \odot$.

Slide 16 of 40

Prefix-Free

Definition

A code $c : S \rightarrow T^*$ is **prefix-free** (short **PF**) if there is no pair of codewords $q = c(s)$ and $q' = c(s')$ such that

$$q' = qr \text{ for some non-empty word } r \in T^*.$$

Example

Let $S = \{w, x, y, z\}$ with $c : S \rightarrow \mathbb{B}^*$ defined by

$$w \mapsto 10 \quad x \mapsto 01 \quad y \mapsto 11 \quad z \mapsto 011$$

This code is not **PF**. It is not **UD** either, because:

$$wxwy \mapsto 10011011 \text{ as well as } wzz \mapsto 10011011$$

Slide 17 of 40

PF implies UD

Theorem

If a code $c : S \rightarrow T^$ is prefix-free then it is uniquely decodable.*

Proof (Indirect).

Take $x = x_1x_2 \cdots x_m$ and $y = y_1y_2 \cdots y_n$ such that $c(x) = c(y)$ with $c(x) = c(x_1)c(x_2) \cdots c(x_m)$ and $c(y) = c(y_1)c(y_2) \cdots c(y_n)$.

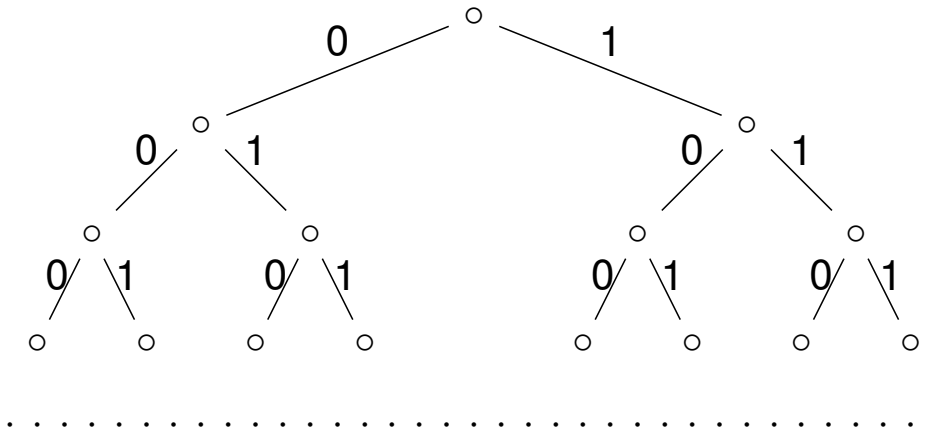
The prefixes (of length k) must be the same $c(x)|_k = c(y)|_k$ for any k . It might be that $|c(x_1)| \neq |c(y_1)|$.

Assume w.l.o.g. $|c(x_1)| \leq |c(y_1)|$, and take $k = |c(x_1)|$ then $c(x)|_k = c(x_1) = c(y_1)|_k$ and thus $c(y_1) = c(x_1)r$ for some $r \in T^*$. This contradicts PF. Repeat this argument to cover all of $c(x) = c(y)$. Formally: Induction on lengths m and n . \square

Slide 18 of 40

Strings as Paths in Trees

If we concentrate on binary codes with $T = \mathbb{B}$ then we can see any possible codeword in C as a (finite) path in a binary tree:



In general, we have to consider a b -ary tree for $|T| = b$.

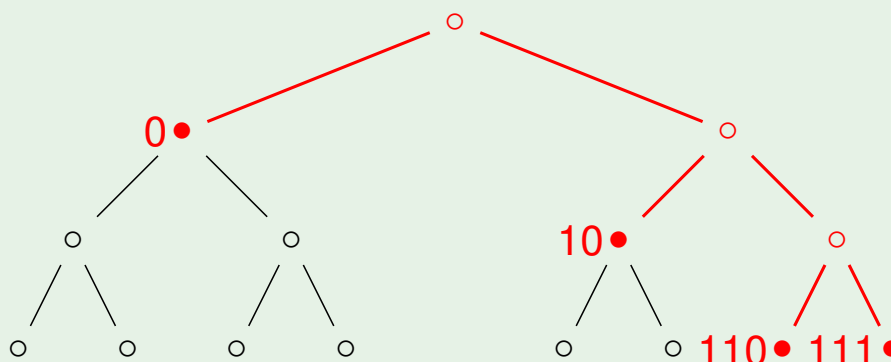
Slide 19 of 40

Codes as (Sub)-Trees

The codewords in C can be represented by the end-points of the corresponding paths. If a code C is PF then none of its descendants can represent a codeword in C .

Example

The code $C = \{0, 10, 110, 111\}$ is PF, its corresponding tree is



Slide 20 of 40

Parameters of a Code

Definition

Given a code $c : S \rightarrow T^*$, let n_i denote the number of symbols $s \in S$ which are decoded by strings of length i , i.e.

$$n_i = |\{s \in S \mid |c(s)| = i\}| = |\{s \in S \mid c(s) \in T^i\}|$$

We refer to n_1, n_2, \dots, n_M as **parameters** of c , where M is the maximal length of a codeword.

The number of all potential codewords of length i is given by $b^i = |T^i|$, in particular: $b = |T| = b^1$.

Note that $\frac{n_i}{b^i}$ gives the fraction of possible codewords of length i which is actually used by a code (“filling rate”).

Slide 21 of 40

Kraft-McMillan Number

Definition

Given a code $c : S \rightarrow T^*$ with parameters n_1, n_2, \dots, n_M . The **Kraft-McMillan number** of c is defined as:

$$K = \sum_{i=1}^M \frac{n_i}{b^i} = \frac{n_1}{b} + \frac{n_2}{b^2} + \dots + \frac{n_M}{b^M}$$

Example

Consider a binary code, i.e. $b = 2$. The parameters of the code are $n_1 = 1, n_2 = 1, n_3 = 2$ (like in the last example), then $b^i = 2^i$, i.e. $b^1 = 2, b^2 = 4, b^3 = 8$ and

$$K = \frac{1}{2} + \frac{1}{4} + \frac{2}{8} = \frac{8}{8} = 1$$

Slide 22 of 40

Existence of PF Codes

Theorem (L.G. Kraft, 1949)

Given an alphabet T with $|T| = b$ and (code) parameters $n_1, n_2, \dots, n_M \in \mathbb{N}$. If $K \leq 1$ there exists a PF b -ary code $c : S \rightarrow T^*$ with these parameters.

Example

Construct a PF binary code with $n_2 = 2, n_3 = 3, n_4 = 1$.

First check that $K = \frac{8}{16} + \frac{6}{16} + \frac{1}{16} = \frac{15}{16} \leq 1$.

- Codewords of length 2: take any two, e.g. **00** and **01**.
- Codewords of length 3: can't take $00\dots$ or $01\dots$ but we can take all of form $1s_2s_3$ with $s_2, s_3 \in \mathbb{B}$; in fact we need only three of **100**, **101**, **110**, and (maybe) not 111.
- Codeword(s) of length 4: Take either **1110** or 1111.

Slide 23 of 40

Proof of Kraft's Theorem

Proof.

Any partial sum of the one defining K also fulfills $\sum_i \frac{n_i}{b^i} \leq 1$, e.g. $\frac{n_1}{b^1} \leq 1$, $\frac{n_1}{b^1} + \frac{n_2}{b^2} \leq 1$, $\frac{n_1}{b^1} + \frac{n_2}{b^2} + \frac{n_3}{b^3} \leq 1$, etc. This implies:

$$n_1 \leq b, \quad n_2 \leq b(b^1 - n_1), \quad n_3 \leq b(b^2 - n_1b^1 - n_2), \quad \text{etc.}$$

or in general:

$$n_i \leq b \left(b^{i-1} - n_1 b^{i-2} - n_2 b^{i-3} - \dots - n_{i-1} \right).$$

- Codeword length 1: Since $n_1 \leq b^1$ we can chose any of them; and $b - n_1$ words of length 1 remain as prefixes for words of length 2, 3, ...

Slide 24 of 40

Proof of Kraft's Theorem (cont.)

Proof (cont).

- Codeword length 2: There are b^2 codewords of length 2 but $b^1 n_1$ are unavailable, so $b^2 - b^1 n_1$ are left. But we have $n_2 \leq b(b - n_1)$ so we can make our choices. Thereafter $b^2 - n_1 b - n_2$ prefixes of length 2 are left.
- Codewords of length i : There are always $b^{i-1} - n_1 b^{i-2} - n_2 b^{i-3} - \dots - n_{i-1}$ prefixes of length $i - 1$ left, and from $K \leq 1$ we have

$$n_i \leq b \left(b^{i-1} - n_1 b^{i-2} - n_2 b^{i-3} - \dots - n_{i-1} \right)$$

So we can choose enough codewords n_i of length i and enough remain.

Formally, by induction on the maximal length of codewords. □

Slide 25 of 40

Encoding Strings

Consider a code $c : S \rightarrow T^*$ with parameters n_1, n_2, \dots, n_M . We aim to encode strings in S of length r . Let $q_r(i)$ be the number of strings of length r in S^* encoded in a string of length i in T^* .

Example

Take $c : S \rightarrow T^*$ with parameters $n_1 = 2, n_2 = 3$; what is $q_2(i)$? Given $x = x_1 x_2 \in S^2$, then $c(x) \in T$ has lengths 2, 3 or 4.

- For $|c(x_1 x_2)| = 2$: Then $|c(x_1)| = 1 = |c(x_2)|$, since $n_1 = 2$ we have $q_2(2) = 2 \times 2 = 4$.
- For $|c(x_1 x_2)| = 3$: Then $|c(x_1)| = 1$ and $|c(x_2)| = 2$ or vice versa, so we get $q_2(3) = 2 \times 3 + 3 \times 2 = 12$.
- For $|c(x_1 x_2)| = 4$: Then $|c(x_1)| = 2 = |c(x_2)|$, thus we know that $q_2(4) = 3 \times 3 = 9$.

Slide 26 of 40

Generating Functions

Generating functions are an important tool for investigating sequences, e.g. enumerations of combinatorial objects.

Definition

For a sequence of numbers $q(1), q(2), q(3), \dots$ the **generating function** $Q(x)$ is given as the polynomial (formal power series) in the unknown variable x :

$$Q(x) = q(1)x + q(2)x^2 + q(3)x^3 + \dots$$

For a code $c : S \rightarrow T^*$ we consider for $1 \leq i \leq rM$:

$$q_r(i) = |\{ |c(s)| = i \mid |s| = r \}|$$

and the generating function:

$$Q_r(x) = q_r(1)x + q_r(2)x^2 + q_r(3)x^3 + \dots + q_r(rM)x^{rM}$$

Slide 27 of 40

Counting Principle (CP)

Theorem

Given a UD code $c : S \rightarrow T^*$ with $|c(s)| \leq M$ for all $s \in S$ with generating function $Q_r(x)$ then for all $r \geq 1$:

$$Q_r(x) = Q_1(x)^r.$$

Example

Take $c : S \rightarrow T^*$ with parameters $n_1 = 2, n_2 = 3$ as above. What is the relationship between Q_1 and Q_2 ?

Clearly $q_1(1) = 2 = n_1$ and $q_1(2) = 3 = n_2$ and thus

$$Q_1(x) = 2x + 3x^2 \quad \text{and} \quad Q_2(x) = 4x^2 + 12x^3 + 9x^4$$

and thus indeed $(Q_1(x))^2 = Q_1(x)Q_1(x) = Q_2(x)$.

Slide 28 of 40

Proof of the Counting Principle

Proof *.

Consider $Q_1(x)^r = (n_1x + n_2x^2 + n_3x^3 + \dots + n_Mx^M)^r$ with $n_j = q_1(j)$. What is the coefficient n to x^k for $1 \leq k \leq rM$? To get any term of the form nx^k multiply r terms from $Q_1(x)$ to get $nx^k = n_{i_1}x^{i_1} \times n_{i_2}x^{i_2} \times \dots \times n_{i_r}x^{i_r}$ with $i_1 + i_2 + \dots + i_r = k$ and then $n = n_{i_1}n_{i_2} \dots n_{i_r}$. The coefficient to x^k is

$$\sum_{i_1+i_2+\dots+i_r=k} n_{i_1}n_{i_2} \dots n_{i_r}$$

Consider $Q_r(x)$ or $q_r(k)$, i.e. the number of words of length k encoded from strings of length r : Each such codeword is formed as $c(x_1)c(x_2) \dots c(x_r)$, we can choose any x_j with $|c(x_j)| = i_j$ as long as $i_1 + i_2 + \dots + i_r = k$ in which case there are $n_{i_1}n_{i_2} \dots n_{i_r}$ possible (input) strings. In total we have to consider again all **partitions** of k □

Slide 29 of 40

Kraft-McMillan Number for UD Codes

Theorem (B. McMillan, 1956)

If there exists a uniquely decodable code $c : S \rightarrow T^$ with parameters $n_1, n_2, \dots, n_M \in \mathbb{N}$ then $K \leq 1$.*

$$\exists \text{ UD} \implies K \leq 1 \implies \exists \text{ PF} \quad \text{and} \quad \text{PF} \implies \text{UD}$$

Corollary

The following statements are equivalent for codes $c : S \rightarrow T^$:*

- *There exists a UD code with parameters n_1, n_2, \dots, n_M .*
- *It holds that $K \leq 1$ for parameters n_1, n_2, \dots, n_M .*
- *There exists a PF code with parameters n_1, n_2, \dots, n_M .*

Slide 30 of 40

Proof of McMillan's Theorem

Proof (Indirect) *.

Fix $r \geq 1$ and take $x = \frac{1}{b}$ in $Q_r(x)$, we get ($M = \max |c(s)|$):

$$Q_r\left(\frac{1}{b}\right) = \frac{q_r(1)}{b} + \frac{q_r(2)}{b^2} + \frac{q_r(3)}{b^3} + \dots + \frac{q_r(rM)}{b^{rM}}.$$

UD implies $q_r(i) \leq b^i$ and thus $\frac{q_r(j)}{b^j} \leq 1$ for all $1 \leq j \leq rM$, and therefore $Q_r\left(\frac{1}{b}\right) \leq rM$. From $K = Q_1\left(\frac{1}{b}\right)$ and by CP we have:

$$K^r = \left(Q_1\left(\frac{1}{b}\right)\right)^r = Q_r\left(\frac{1}{b}\right) \leq rM \quad \text{thus} \quad \frac{K^r}{r} \leq M \quad \forall r.$$

Suppose $K > 1$ or $K = 1 + k$ with $k > 0$, then $K^r = (1 + k)^r = 1 + rk + \frac{1}{2}r(r-1)k^2 + \dots$. Furthermore thus $K^r > \frac{1}{2}r(r-1)k^2$ or $\frac{K^r}{r} > \frac{1}{2}(r-1)k^2$ which, for large r , contradicts $\frac{K^r}{r} \leq M$.



Slide 31 of 40

Some Probability Theory

Probability theory is concerned with quantifying or measuring the chances that certain **events** can happen.

We consider an **event space**, i.e. a finite set Ω and a set $\mathcal{B} \subseteq \mathcal{P}(\Omega)$ of **measurable** sets in Ω which form a Boolean algebra (based on union, intersection and complement). For finite sets one can use the power-set $\mathcal{B} = \mathcal{P}(\Omega)$.

Probabilities are then assigned via a **measure**, i.e. a function $\Pr : \mathcal{B} \rightarrow \mathbb{R}$ or $m : \mathcal{B} \rightarrow \mathbb{R}$ or $\mu : \mathcal{B} \rightarrow \mathbb{R}$.

Note: For (uncountable) infinite Ω one needs to develop a more general **measure theory** (e.g. with \mathcal{B} a σ -algebra) to overcome problems like $\mu([0, 1]) = 1$ but $\mu(x) = 0$ for all $x \in [0, 1]$.

Finite Probability Spaces

Consider a finite **measurable spaces** (Ω, \mathcal{B}) with $|\Omega| = n$,

Definition

A **probability** (measure) $\Pr : \mathcal{B}$ on (Ω, \mathcal{B}) has to fulfill

- $\Pr(\Omega) = 1$.
- $0 \leq \Pr(A) \leq 1$ for all $A \in \mathcal{B}$.
- $\Pr(A \cup B) = \Pr(A) + \Pr(B)$ for $A \cap B = \emptyset$.

Some further rules (which follow from the axioms above):

- $\Pr(\emptyset) = 0$,
- $\Pr(\bar{A}) = 1 - \Pr(A)$,
- $\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$.
- etc.

Slide 33 of 40

Random Distributions

For finite **probability spaces** $(\Omega, \mathcal{B}, \Pr)$ with $|\Omega| = n$, we can define a probability (measure) via atoms in $\omega \in \Omega$.

Definition

A **probability distribution** is a function $\mathbf{p} : \Omega \rightarrow [0, 1]$, with

$$\sum_{\omega \in \Omega} \mathbf{p}(\omega) = 1.$$

If we enumerate the elements in Ω in some arbitrary way as $\Omega = \{\omega_1, \omega_2, \dots, \omega_n\}$ then we can also represent \mathbf{p} by a (row) vector in \mathbb{R}^n .

$$\mathbf{p} = (\mathbf{p}(\omega_1), \mathbf{p}(\omega_2), \dots, \mathbf{p}(\omega_n))$$

Slide 34 of 40

Random Variables

Probability distributions on a finite Ω define a probability (measure) in the obvious way

$$\Pr(A) = \sum_{\omega \in A} \mathbf{p}(\omega).$$

Definition

A **random variable** is function $X : \Omega \rightarrow \mathbb{R}$.

We can represent random variables as (column) vectors in \mathbb{R}^n .

Example

Consider a dice. The event space describes the top face of the dice, i.e. $\Omega = \left\{ \begin{array}{|c|} \hline \bullet \\ \hline \end{array}, \begin{array}{|c|} \hline \bullet \bullet \\ \hline \end{array}, \begin{array}{|c|} \hline \bullet \bullet \bullet \\ \hline \end{array}, \begin{array}{|c|} \hline \bullet \bullet \bullet \bullet \\ \hline \end{array}, \begin{array}{|c|} \hline \bullet \bullet \bullet \bullet \bullet \\ \hline \end{array}, \begin{array}{|c|} \hline \bullet \bullet \bullet \bullet \bullet \bullet \\ \hline \end{array} \right\}$, define X which counts the number of eyes, e.g. $X \left(\begin{array}{|c|} \hline \bullet \bullet \bullet \bullet \bullet \\ \hline \end{array} \right) = 5$ etc.

Slide 35 of 40

Moments in Probability

Expectation Value. For a random variable X and probability distribution \mathbf{p} we define:

$$\mathbf{E}(X) = \sum_{\omega \in \Omega} \mathbf{p}(\omega) X(\omega) = \sum_i \mathbf{p}_i \mathbf{X}_i = \mu_X$$

One can show: $\mathbf{E}(X + Y) = \mathbf{E}(X) + \mathbf{E}(Y)$ and $\mathbf{E}(\alpha X) = \alpha \mathbf{E}(X)$.

Example

Consider a (fair) dice and previous random variable X , then

$$\mathbf{E}(X) = 1 \frac{1}{6} + 2 \frac{1}{6} + 3 \frac{1}{6} + 4 \frac{1}{6} + 5 \frac{1}{6} + 6 \frac{1}{6} = \frac{21}{6}$$

Variance. For random variable X and distribution \mathbf{p} we define:

$$\text{Var}(X) = \mathbf{E}((X - \mathbf{E}(X))^2) = \mathbf{E}(X^2) - (\mathbf{E}(X))^2.$$

The **standard deviation** is $\sigma_X = \sqrt{\text{Var}(X)}$.

Slide 36 of 40

Bayes Theorem and Independence

Given two subsets A and B in a probability space $(\Omega, \mathcal{B}, \Pr)$. The **conditional probability** of A given that B has happened is

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

One can show **Bayes Theorem** which states:

$$\Pr_A(B) = \frac{\Pr_B(A)\Pr(B)}{\Pr(A)} = \frac{\Pr(A | B)\Pr(B)}{\Pr(A)} = \Pr(B | A).$$

Given two subsets A and B in a probability space $(\Omega, \mathcal{B}, \Pr)$, A and B are (probabilistically) **independent** if

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

Slide 37 of 40

Products and Probability

Given two probability spaces (Ω_1, \Pr_1) and (Ω_2, \Pr_2) , to keep things simple use $\mathcal{B}_i = \mathcal{P}(\Omega_i)$. We can define a probability \Pr on the cartesian product $\Omega = \Omega_1 \times \Omega_2$ via:

$$\Pr(\langle \omega_1, \omega_2 \rangle) = \Pr_1(\omega_1)\Pr_2(\omega_2)$$

If \Pr_1 and \Pr_2 correspond to probability distributions \mathbf{p}_1 and \mathbf{p}_2 and \mathbf{p} to \Pr then $\mathbf{p} = \mathbf{p}_1 \otimes \mathbf{p}_2$, i.e. the **tensor** product.

Caveat: Not all distributions on $\Omega_1 \times \Omega_2$ are a product.

Example

Consider $\Omega_1 = \{0, 1\}$ and $\Omega_2 = \{z, o\}$ and probability $\Pr(\langle 0, z \rangle) = \Pr(\langle 1, o \rangle) = \frac{1}{2}$ and $\Pr(\langle 0, o \rangle) = \Pr(\langle 1, z \rangle) = 0$ **cannot** be represented as a product $\mathbf{p}_1 \otimes \mathbf{p}_2$. However, one can show: $\mathbf{p} = \frac{1}{2}(1, 0) \otimes (1, 0) + \frac{1}{2}(0, 1) \otimes (0, 1)$.

Slide 38 of 40

Tensor/Kronecker Product

Given a $n \times m$ matrix \mathbf{A} and a $k \times l$ matrix \mathbf{B} :

$$\mathbf{A} = \begin{pmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nm} \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} b_{11} & \dots & b_{1l} \\ \vdots & \ddots & \vdots \\ b_{k1} & \dots & b_{kl} \end{pmatrix}$$

The **tensor** or **Kronecker product** $\mathbf{A} \otimes \mathbf{B}$ is a $nk \times ml$ matrix:

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & \dots & a_{1m}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{n1}\mathbf{B} & \dots & a_{nm}\mathbf{B} \end{pmatrix}$$

Special cases are square matrices ($n = m$ and $k = l$) and **vectors** (row $n = k = 1$, column $m = l = 1$).

Slide 39 of 40

Correlation

The **covariance** of two random variables X and Y is:

$$\text{Cov}(X, Y) = \mathbf{E}((X - \mathbf{E}(X))(Y - \mathbf{E}(Y))) = \mathbf{E}(XY) - \mathbf{E}(X)\mathbf{E}(Y)$$

The **correlation coefficient** is $\rho(X, Y) = \text{Cov}(X, Y)/(\sigma_X\sigma_Y)$.

For **independent** random variables X and Y – i.e. if we have $\Pr((X = x_j) \cap (Y = y_k)) = \Pr(X = x_j)\Pr(Y = y_k)$:

$$\mathbf{E}(XY) = \mathbf{E}(X)\mathbf{E}(Y) \text{ and } \text{Cov}(X, Y) = \rho(X, Y) = 0.$$

Note that $\rho(X, Y) = 0$ does **not** imply independence.

Example

$\Pr(X = x) = \frac{1}{3}$ for $x = -1, 0, 1$ and $Y = X^2$, then $\rho(X, Y) = 0$.

Slide 40 of 40