Machine Learning
Machine Learning: what is it?

We have data

\[0 \, 1 \, 2 \, 3 \, 4\]

\[5 \, 6 \, 7 \, 8 \, 9\]
Machine Learning: what is it?

We have data

We want

...a model

...a prediction

0  1  2  3  4  5  6  7  8  9

Classification

Regression
Machine Learning: what is it?

We have data

We want

...a model

...a prediction

In machine learning we use generic models that we train from the data.
Machine Learning: what is it?

We have data...a model...a prediction

In machine learning we use **generic models** that we train from the data.

Prediction is achieved using the **trained models**.
Crash course in probability theory and statistics
Motivation

**Problem:** To avoid relying on “magic” we need mathematics. For machine learning we need to quantify:

- Uncertainty in data measures and conclusions
- “Goodness” of model (when confronted with data)
- Expected error and expected success rates
- ...and many similar quantities...
Motivation

**Problem:** To avoid relying on “magic” we need mathematics. For machine learning we need to quantify:
- Uncertainty in data measures and conclusions
- “Goodness” of model (when confronted with data)
- Expected error and expected success rates
- ...and many similar quantities...

**Probability theory:** Mathematical modelling when uncertainty or randomness is present.
Motivation

**Problem:** To avoid relying on “magic” we need mathematics. For machine learning we need to quantify:
- Uncertainty in data measures and conclusions
- “Goodness” of model (when confronted with data)
- Expected error and expected success rates
- ...and many similar quantities...

**Probability theory:** Mathematical modelling when uncertainty or randomness is present.

**Statistics:** The mathematics of collection of data, description of data, and inference from data
ACME Algorithm

• I have a linear time algorithm
• Need to handle size $1e4$ in reasonable time (say a week)
• Can I do that?
Input size versus running time

Input size (n)

Running time (t)
A *discrete random variable*, \( X \), is a variable that can take values in a discrete (countable) set \( \{x_i\} \).

The *probability* of \( X \) taking the value \( x_i \) is denoted \( p(X=x_i) \).

\[
P(X = x_i) \geq 0
\]

\[
\sum_i P(X = x_i) = 1
\]
Continuous random variables

A **continuous random variable**, $X$, is a variable that can take values in $\mathbb{R}^d$.

The **probability density** of $X$ is an integrable function $p(X)$

\[
p(x) \geq 0 \quad \int p(x) \, dx = 1
\]

The **probability** of $X \in S \subseteq \mathbb{R}^d$ is given by

\[
p(x \in S) = \int_S p(x) \, dx
\]
A model of the time usage?
Deterministic part and random part...

Randomness!
Input size versus running time

\[ t = \beta n + \alpha + \epsilon \]

Randomness!
Parameterized distributions

Many distributions are governed by a few parameters.

E.g. coin tossing (Bernoulli distribution) governed by the probability of “heads”.

Binomial distribution: number of “heads” $k$ out of $n$ coin tosses:

$$p(k \mid n, \theta) \propto \theta^k (1 - \theta)^{n-k}$$

$$p(\text{head} \mid \theta) = \theta$$

$$p(\text{tail} \mid \theta) = 1 - \theta$$
Input size versus running time

\[ t = \beta n + \alpha + \epsilon \]

Randomness!
\[ t = \beta n + \alpha + \epsilon \]

\[ \epsilon \sim N(0, \sigma) \]
Gaussian/Normal distribution

\[ N(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{1}{2\sigma^2} (x - \mu)^2 \right\} \]
Input size versus running time

\[ t = \beta n + \alpha + \epsilon \]

Randomness!

\[ \epsilon \sim N(0, \sigma) \]

\[ t \sim N(\beta n + \alpha, \sigma) \]
Input size versus running time

$t = \beta n + \alpha$
Input size versus running time

Parameters:
\[ t = \beta n + \alpha \]
Assume $D = \{x_1, x_2, ..., x_N\}$ are independent, identically distributed (i.i.d) outcomes of our experiments (observed data).

Desirable properties of an estimator are:

$$
\hat{\mu} \to \mu \text{ for } N \to \infty \quad \text{(consistent)}
$$

and

$$
\mathbb{E}[\hat{\mu}] = \mu \quad \text{(unbiased)}
$$
A general way to get an estimator is using the maximum likelihood (ML) or maximum a posterior (MAP) approach:

**ML:**
\[
\hat{\mu} = \maxarg_{\mu} p(x \mid \mu)
\]

**MAP:**
\[
\hat{\mu} = \maxarg_{\mu} p(\mu \mid x) = \maxarg_{\mu} p(x \mid \mu)p(\mu)
\]

...possibly compensating for any bias in these.
The function $x \rightarrow p(x \mid q)$ is the *probability* of observation $x$ given parameter $q$.

The function $q \rightarrow p(x \mid q)$ is the *likelihood* of parameter $q$ given observation $x$. Sometimes written $\text{lhd}(q \mid x) = p(x \mid q)$. 

**Parameterized distributions**
Parameter estimation

**Maximum Likelihood (ML):**

**Maximum A Posteriori (MAP):**
(A Bayesian approach assuming a distribution over parameters).

**Fully Bayesian:**
(Estimates a distribution rather than a parameter).

\[
\hat{\theta} = \maxarg_{\theta} p(x | \theta) \\
\hat{\theta} = \maxarg_{\theta} p(\theta | x) \\
= \maxarg_{\theta} p(x | \theta)p(\theta) \\
\hat{p}(y | x) = \int p(y | \theta)p(\theta | x) \, d\theta \\
\propto \int p(y | \theta)p(x | \theta)p(\theta) \, d\theta
\]
Parameter estimation

**Maximum Likelihood (ML):**

\[
\hat{\theta} = \max_{\theta} p(x | \theta)
\]

**Maximum A Posteriori (MAP):**

(A Bayesian approach assuming a distribution over parameters).

\[
\hat{\theta} = \max_{\theta} p(\theta | x) = \max_{\theta} p(x | \theta)p(\theta)
\]

**Fully Bayesian:**

(Estimates a distribution rather than a parameter).

\[
\hat{p}(y | x) = \int p(y | \theta)p(\theta | x) \, d\theta
\]

\[
\propto \int p(y | \theta)p(x | \theta)p(\theta) \, d\theta
\]
Input size versus running time

\[ t = \beta n + \alpha \]
Predictions

When observing new $n$ we can use $p(t \mid n, q)$ to make predictions about $t$. 
Input size versus running time

\[ t = \beta n + \alpha \]
Input size versus running time

\[ t = \beta n + \alpha \]

95% confidence interval

Capturing uncertainty
Parameter estimation

**Maximum Likelihood (ML):**

\[
\hat{\theta} = \max_{\theta} p(x \mid \theta)
\]

**Maximum A Posteriori (MAP):**
(A Bayesian approach assuming a distribution over parameters).

\[
\hat{\theta} = \max_{\theta} p(\theta \mid x) = \max_{\theta} p(x \mid \theta)p(\theta)
\]

**Fully Bayesian:**
(Estimates a distribution rather than a parameter).

\[
\hat{p}(y \mid x) = \int p(y \mid \theta)p(\theta \mid x) \, d\theta \\
\propto \int p(y \mid \theta)p(x \mid \theta)p(\theta) \, d\theta
\]
Joint probability

If a random variable, $Z$, is a vector, $Z=(X,Y)$, we can consider its components separately.

The probability $p(Z=z)$ where $z = (x,y)$ is the joint probability of $X=x$ and $Y=y$ written $p(X=x,Y=y)$ or $p(x,y)$.

When clear from context, we write just $p(X,Y)$ or $p(x,y)$ and the notation is symmetric: $p(X,Y) = p(Y,X)$ and $p(x,y) = p(y,x)$. 
Marginal probability

The sum rule:

\[ p(X) = \sum_Y p(X, Y) \]
Conditional probability

The **conditional probability of X given Y** is written $P(X|Y)$ and is the quantity satisfying $p(X,Y) = p(X|Y)p(Y)$.

**The product rule:**

$$p(X, Y) = p(X|Y)P(Y)$$

When $p(Y) \neq 0$ we get $p(X|Y) = p(X,Y) / p(Y)$ with a simple interpretation.
Conditional probability

The *conditional probability of X given Y* is written $P(X|Y)$ and is the quantity satisfying $p(X,Y) = p(X|Y)p(Y)$.

**Intuition:** Before we observe anything, the probability of $X$ is $p(X)$ but after we observe $Y$ it becomes $p(X|Y)$.

$$p(X, Y) = p(X|Y)P(Y)$$

When $p(Y) \neq 0$ we get $p(X|Y) = p(X,Y) / p(Y)$ with a simple interpretation.
Independence

When \( p(X,Y) = p(X)p(Y) \) we say that \( X \) and \( Y \) are independent.

In this case:

\[
p(X | Y) = \frac{p(X,Y)}{p(Y)} = \frac{p(X)p(Y)}{p(Y)} = p(X)
\]

**Intuition/justification:** Observing \( Y \) does not change the probability of \( X \).
Bayes' theorem

Since $p(X,Y) = p(Y,X)$ (symmetry) and $p(X,Y) = p(Y|X)p(X)$ (product rule) it follows $p(Y|X)p(X) = p(X|Y)p(Y)$ or, when $p(X) \neq 0$:

**Bayes' theorem:**

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

Sometimes written: $p(Y|X) \propto p(X|Y)p(Y)$ where $p(X) = \sum_Y p(X|Y)p(Y)$ is an implicit *normalising factor*. 
Parameter estimation

**Example:** We toss a coin and get a “head”. Our model is a binomial distribution; $x$ is one “head” and $q$ the probability of a “head”.

Likelihood: 
$$lhd(\theta \mid x) = p(x \mid \theta) \propto \theta^1 (1 - \theta)^0$$

Prior: 
$$p(\theta) \propto \theta^\alpha (1 - \theta)^\beta$$

Posterior: 
$$p(\theta \mid x) \propto \theta^{\alpha + 1} (1 - \theta)^\beta$$
Example: We toss a coin and get a “head”. Our model is a binomial distribution; \( x \) is one “head” and \( q \) the probability of a “head”.

Likelihood:

\[
\text{ld}(\theta \mid x) = p(x \mid \theta) \propto \theta^1 (1 - \theta)^0
\]

Prior:

\[
p(\theta) \propto \theta^\alpha (1 - \theta)^\beta
\]

Posterior:

\[
p(\theta \mid x) \propto \theta^{\alpha + 1} (1 - \theta)^\beta
\]
Parameter estimation

**Example:** We toss a coin and get a “head”. Our model is a binomial distribution; \(x\) is one “head” and \(q\) the probability of a “head”.

Likelihood: \[\text{ld}(\theta \mid x) = p(x \mid \theta) \propto \theta^1(1 - \theta)^0\]

Prior: \[p(\theta) \propto \theta^\alpha(1 - \theta)^\beta\]

Posterior: \[p(\theta \mid x) \propto \theta^{\alpha+1}(1 - \theta)^0\]

MAP estimate
Parameter estimation

**Example:** We toss a coin and get a “head”. Our model is a binomial distribution; $x$ is one “head” and $q$ the probability of a “head”.

**Fully Bayesian approach:**

\[
\hat{p}(y \mid x) = \int_0^1 p(y \mid x) p(\theta \mid x) \, d\theta \\
\propto \int_0^1 \theta^{\alpha+1} (1 - \theta)^{\beta+1 - 1_{y=h}} \, d\theta \\
= B(\alpha + 1 + 1_{y=h}, \beta + 1 - 1_{y=h})
\]

**Posterior:**

\[
p(\theta \mid x) \propto \theta^{\alpha+1} (1 - \theta)^{\beta}
\]
Bayesian estimation

In a fully Bayesian approach we instead update our distribution based on observed data:

\[
\hat{p}(y \mid x) = \int p(y \mid \mu)p(\mu \mid x) \, d\mu \\
\propto \int p(y \mid \mu)p(x \mid \mu)p(\mu) \, d\mu
\]
Frequencist approach

Input size (n)

Running time (t)

Baysian approach

Input size (n)

Running time (t)
Frequencist approach

Baysian approach
Model selection

Where do we get $p(t, x \mid q)$ from in the first place?
Where do we get $p(t, x | q)$ from in the first place?

There is no *right* model – fair dice is as unrealistic as a spherical cow!
Where do we get $p(t,x | q)$ from in the first place?

There is no *right* model – fair dice is as unrealistic as a spherical cow!

Sometimes there are obvious candidates to try – either for the joint or conditional probabilities $p(x,t | q)$ or $p(t | x, q)$.

Sometimes we can try a "generic" model – linear models, neural networks, ...
Model selection

Where do we get $p(t,x \mid q)$ from in the first place?

There is no right model – fair dice is as unrealistic as a spherical cow!

Sometimes there are obvious candidates to try – either for the joint or conditional probabilities $p(x,t \mid q)$ or $p(t \mid x,q)$.

Sometimes we can try a "generic" model – linear models, neural networks, ...

This is the topic of most of this class!
Model selection

Where do we get $p(t,x \mid q)$ from in the first place?

There is no *right* model – fair dice is as unrealistic as a spherical cow!

But some models are more *useful* than others.
Where do we get $p(t,x \mid q)$ from in the first place?

There is no *right* model – fair dice is as unrealistic as a spherical cow!

But some models are more *useful* than others.

If we have several models, how do we measure the usefulness of each?
Model selection

Where do we get $p(t,x \mid q)$ from in the first place?

There is no **right** model – fair dice is as unrealistic as a spherical cow!

But some models are more **useful** than others.

If we have several models, how do we measure the usefulness of each?

A good measure is prediction accuracy on new data.
If we compare two models, we can take a maximum likelihood approach:

$$M = \arg\max_M p(t, x \mid M)$$

or a Bayesian approach:

$$M = \arg\max_M p(t, x \mid M)p(M)$$

just as for parameters.
Model selection

If we compare two models, we can take a maximum likelihood approach:

Or a Bayesian approach:

But there is an overfitting problem:

Complex models often fit training data better without generalizing better!
Model selection

If we compare two models, we can take a maximum likelihood approach:

But there is an over fitting problem:

Complex models often fit training data better without generalizing better!

In Bayesian approach, use $p(M)$ to penalize complex models

In ML approach, use some Information Criteria and maximize $\ln p(t,x | M) - \text{penalty}(M)$. 
Model selection

If we compare two models, we can take a maximum likelihood approach:

But there is an over fitting problem:

Complex models often fit training data better without generalizing better!

Or more empirical approach: Use some method of splitting data into training data and test data and pick model that performs best on test data.

(and retrain that model with the full dataset).
Decision theory

Based on $p(x,t \mid q)$ we often need to make decisions.

This often means taking one of a small set of actions $A_1,A_2,\ldots,A_k$ based on observed $x$.

Assume that the target variable is in this set, then we make decisions based on $p(t \mid x, q)$ or $p(A_i \mid x, q)$.

Put in a different way: we use $p(x,t \mid q)$ to classify $x$ into one of $k$ classes, $C_i$. 
Decision theory

We can approach this by splitting the input into regions.

In $R_1$ go for $C_1$; in $R_2$ go for $C_2$.

Choose regions to minimize classification errors:

$$p(\text{mistake}) = p(x \in R_1, C_2) + p(x \in R_2, C_1)$$

$$= \int_{R_1} p(x, C_2) \, dx + \int_{R_2} p(x, C_1) \, dx$$
We can approach this by splitting the input into regions

In $R_1$ go for $C_1$; in $R_2$ go for $C_2$.

Choose regions to minimize classification errors:

$$p(\text{mistake}) = p(x \in R_1, C_2) + p(x \in R_2, C_1)$$

$$= \int_{R_1} p(x, C_2) \, dx + \int_{R_2} p(x, C_1) \, dx$$

Red and green mis-classifies $C_2$ as $C_1$

Blue mis-classifies $C_1$ as $C_2$

At $x_0$ red is gone and $p(\text{mistake})$ is minimized
String classification

• You have a note illustrating most of these ideas in terms of string classification
• Do the exercises for next week; we will spend the first half of next week's lecture going through the exercises
  - Bring your laptop with your code for the exercises!