

# **Generalisation and Likelihoods**

**CS4061 / CS5014 Machine Learning**

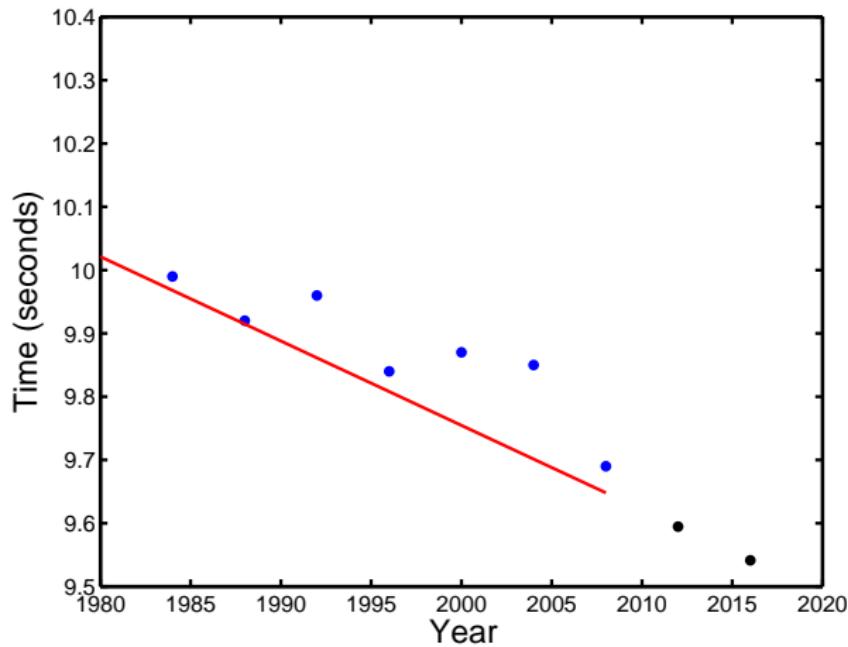
**Paul Henderson**

University of Glasgow

`paul.henderson@glasgow.ac.uk`

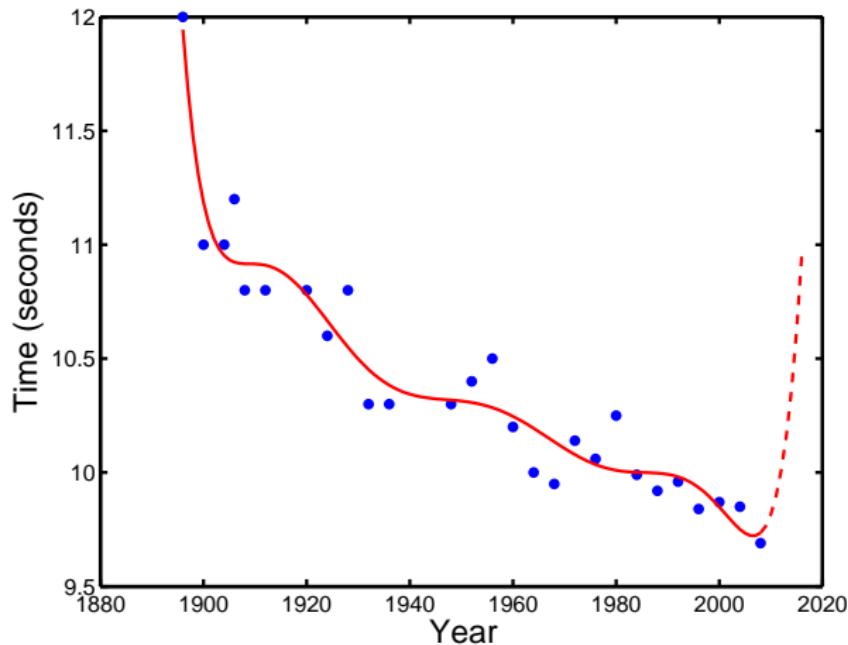
*Based on previous material by  
Simon Rogers & Ke Yuan*

# Olympic data



Linear model – predictions OK?

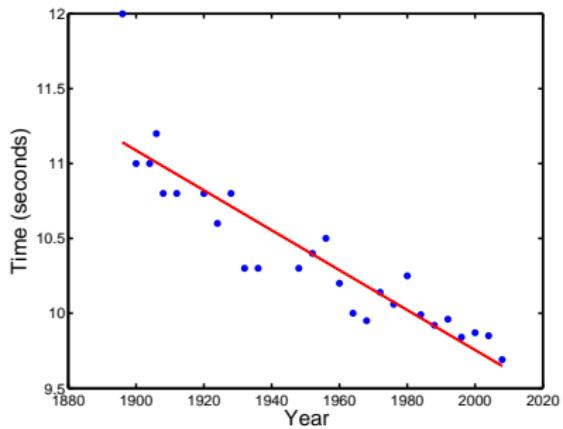
# Olympic data



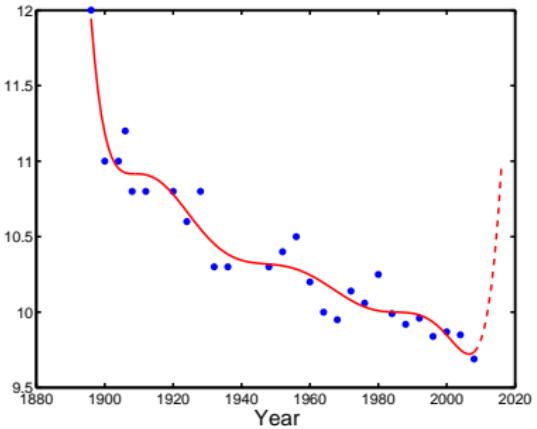
8<sup>th</sup> order model – predictions terrible!

**generalisation** is important

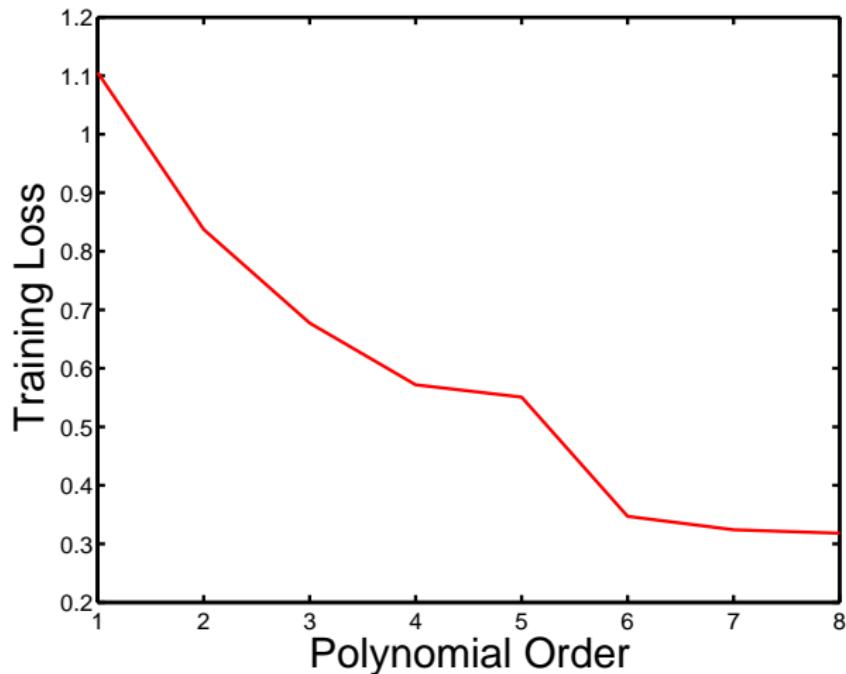
# How to choose?



vs.

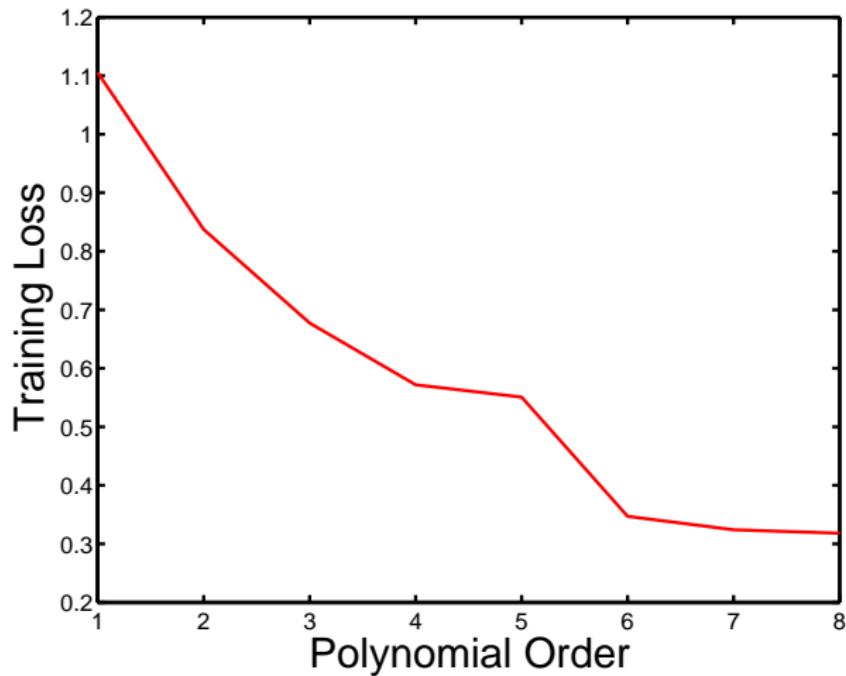


## How does loss change?



Loss,  $\mathcal{L}$ , on Olympic data as terms ( $x^k$ ) are added to the model

## How does loss change?

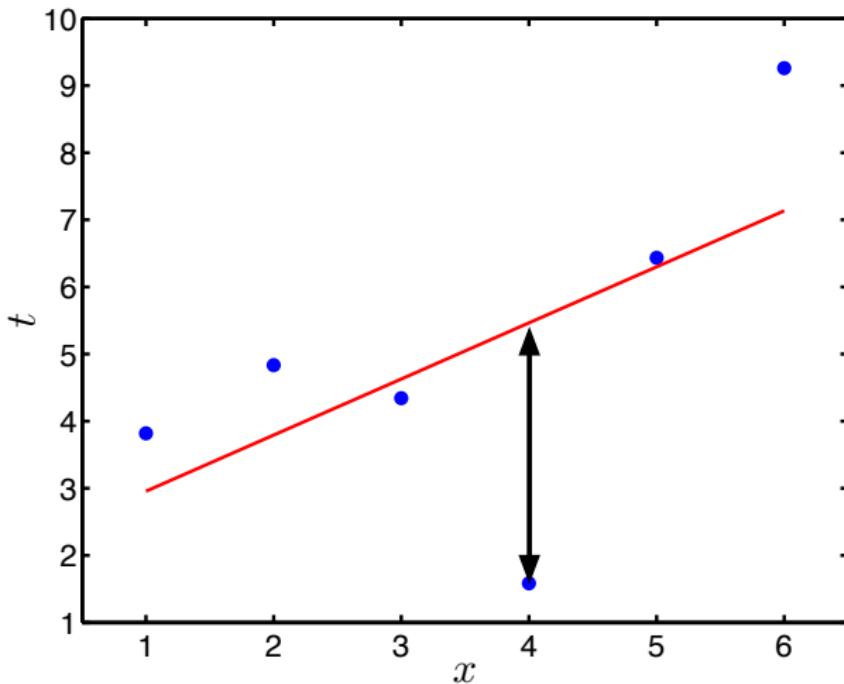


Loss,  $\mathcal{L}$ , on Olympic data as terms ( $x^k$ ) are added to the model

Loss **always** decreases as the model is made more complex

## Loss always decreases with model complexity

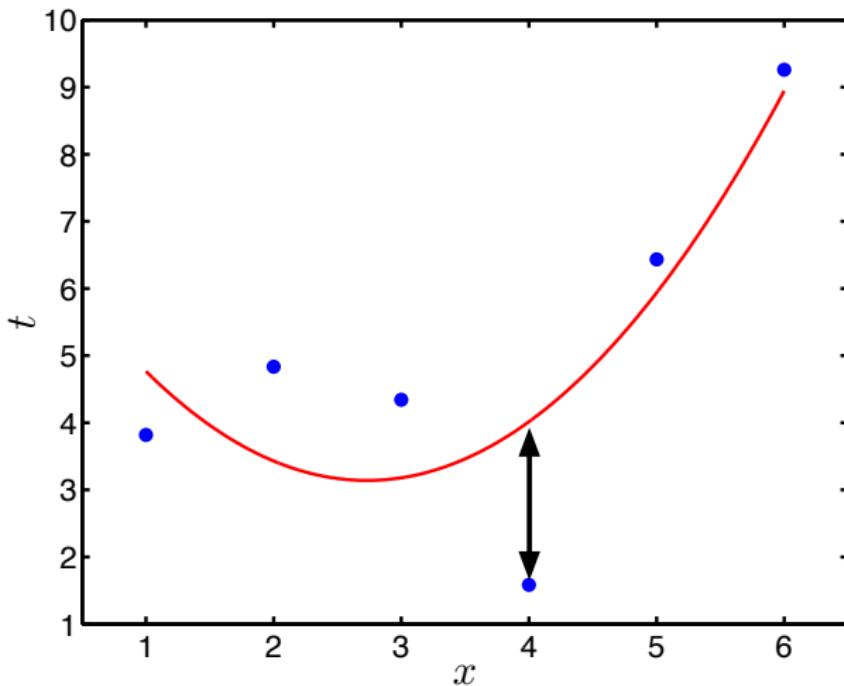
Data comes from  $t = x$  with some *noise* added:



Linear model  $t = w_0 + w_1x$ .

## Loss always decreases with model complexity

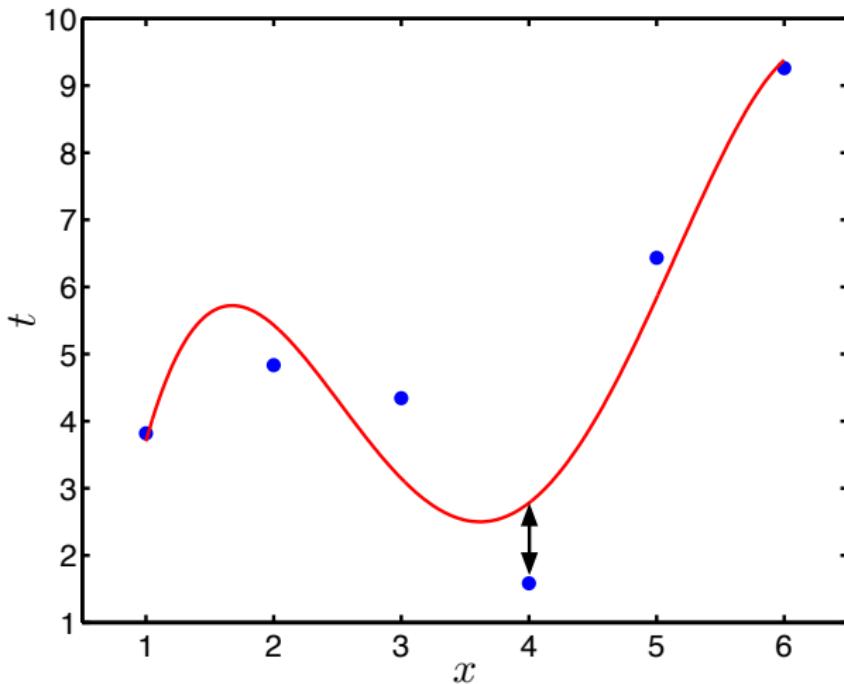
Data comes from  $t = x$  with some *noise* added:



Quadratic model  $t = w_0 + w_1x + w_2x^2$ .

## Loss always decreases with model complexity

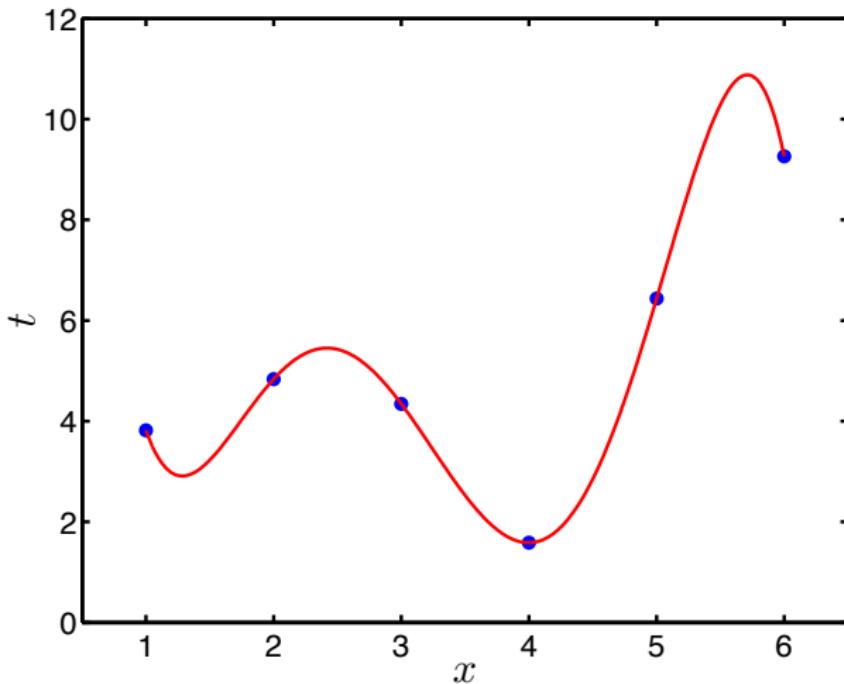
Data comes from  $t = x$  with some *noise* added:



$$\text{Fourth order } t = w_0 + w_1x + w_2x^2 + w_3x^3 + w_4x^4.$$

## Loss always decreases with model complexity

Data comes from  $t = x$  with some *noise* added:



$$\text{Fifth order } t = w_0 + w_1x + w_2x^2 + w_3x^3 + w_4x^4 + w_5x^5.$$

# Generalisation and over-fitting

Trade-off: **generalisation** (predictive ability)  
vs. **over-fitting** (decreasing the loss)

# Generalisation and over-fitting

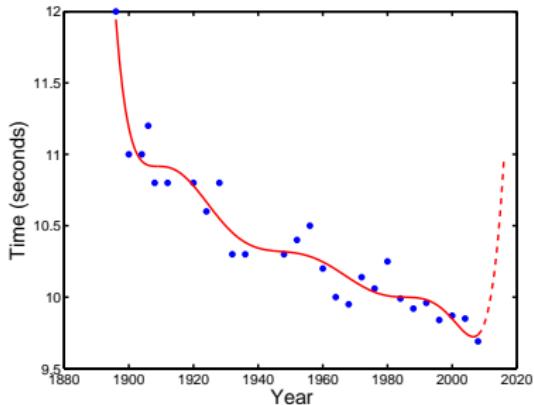
Trade-off: **generalisation** (predictive ability)  
vs. **over-fitting** (decreasing the loss)

- ▶ *Perfectly* fitting the training data typically leads to poor predictions
- ▶ The model will have fitted whatever *noise* is present

# Generalisation and over-fitting

Trade-off: **generalisation** (predictive ability)  
vs. **over-fitting** (decreasing the loss)

- ▶ *Perfectly* fitting the training data typically leads to poor predictions
- ▶ The model will have fitted whatever *noise* is present



## Noise

Not necessarily ‘noise’, just things we can’t, or don’t need to model

## Possible ways of choosing

- ▶ Lowest loss,  $\mathcal{L}$ ?
  - ▶ Loss always decreases as model gets more complex

## Possible ways of choosing

- ▶ Lowest loss,  $\mathcal{L}$ ?
  - ▶ Loss always decreases as model gets more complex
  - ▶ Predictions don't necessarily get better

## Possible ways of choosing

- ▶ Lowest loss,  $\mathcal{L}$ ?
  - ▶ Loss always decreases as model gets more complex
  - ▶ Predictions don't necessarily get better
- ▶ Best predictions?
  - ▶ On what data?

## Where can we get more data?

- ▶ We have  $N$  input-response pairs for training:

$$(x_1, t_1), (x_2, t_2), \dots, (x_N, t_N).$$

## Where can we get more data?

- ▶ We have  $N$  input-response pairs for training:

$$(x_1, t_1), (x_2, t_2), \dots, (x_N, t_N).$$

- ▶ We could use  $N - C$  pairs to find  $\hat{\mathbf{w}}$  for several models.

## Where can we get more data?

- ▶ We have  $N$  input-response pairs for training:

$$(x_1, t_1), (x_2, t_2), \dots, (x_N, t_N).$$

- ▶ We could use  $N - C$  pairs to find  $\hat{\mathbf{w}}$  for several models.
- ▶ Choose the model that makes best predictions on remaining  $C$  pairs.

## Where can we get more data?

- ▶ We have  $N$  input-response pairs for training:

$$(x_1, t_1), (x_2, t_2), \dots, (x_N, t_N).$$

- ▶ We could use  $N - C$  pairs to find  $\hat{\mathbf{w}}$  for several models.
- ▶ Choose the model that makes best predictions on remaining  $C$  pairs.
  - ▶ The  $N - C$  pairs constitute *training data*.
  - ▶ The  $C$  pairs are known as *validation data*.

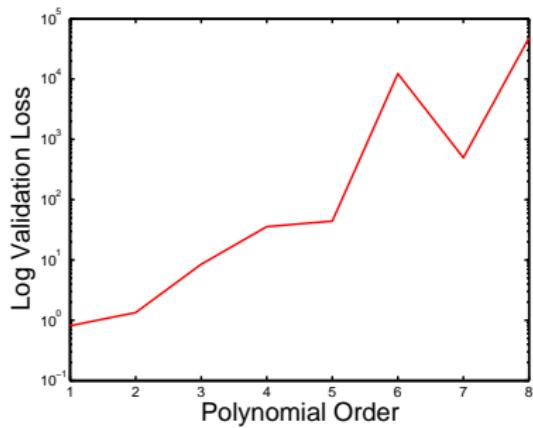
## Where can we get more data?

- ▶ We have  $N$  input-response pairs for training:

$$(x_1, t_1), (x_2, t_2), \dots, (x_N, t_N).$$

- ▶ We could use  $N - C$  pairs to find  $\hat{\mathbf{w}}$  for several models.
- ▶ Choose the model that makes best predictions on remaining  $C$  pairs.
  - ▶ The  $N - C$  pairs constitute *training data*.
  - ▶ The  $C$  pairs are known as *validation data*.
- ▶ Example – use Olympics pre 1980 to train and post 1980 to validate.

# Validation example



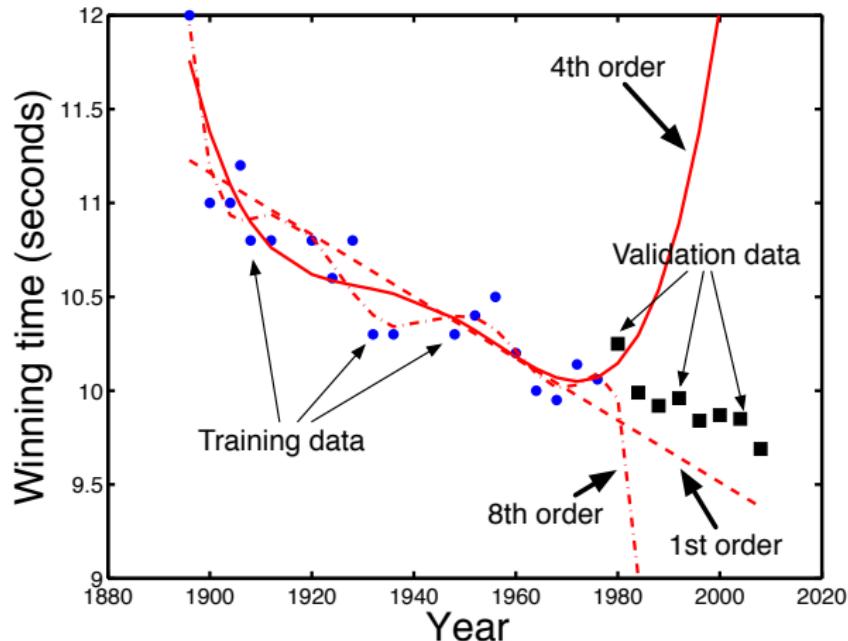
Predictions evaluated using validation loss:

$$\mathcal{L}_v = \frac{1}{C} \sum_{c=1}^C (t_c - \mathbf{w}^\top \mathbf{x}_c)^2$$

Best model?

Results suggest that a first order (linear) model ( $t = w_0 + w_1x$ ) is best.

## Validation example



Best model

First order (linear) model generalises best.

## How should we choose which data to hold back?

- ▶ In some applications it will be clear.
  - ▶ Olympic data – validating on the most recent data seems sensible.
- ▶ In many cases – pick it randomly.

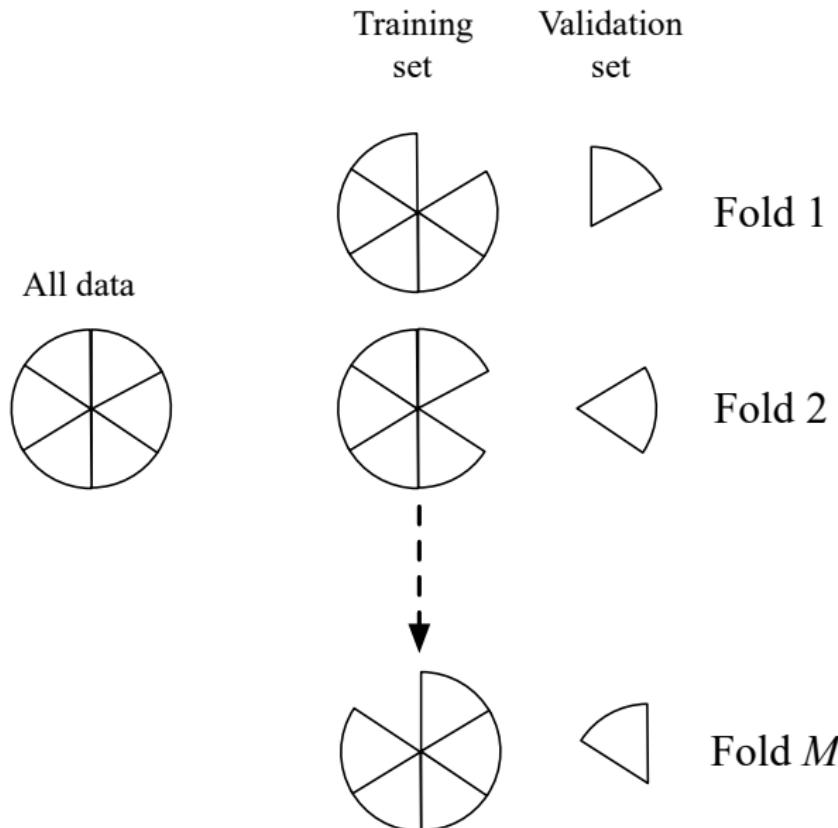
## How should we choose which data to hold back?

- ▶ In some applications it will be clear.
  - ▶ Olympic data – validating on the most recent data seems sensible.
- ▶ In many cases – pick it randomly.
- ▶ Do it more than once – average the results.

## How should we choose which data to hold back?

- ▶ In some applications it will be clear.
  - ▶ Olympic data – validating on the most recent data seems sensible.
- ▶ In many cases – pick it randomly.
- ▶ Do it more than once – average the results.
- ▶ Do cross-validation.
  - ▶ Split the data into  $M$  equal sets. Train on  $M - 1$ , test on remaining.

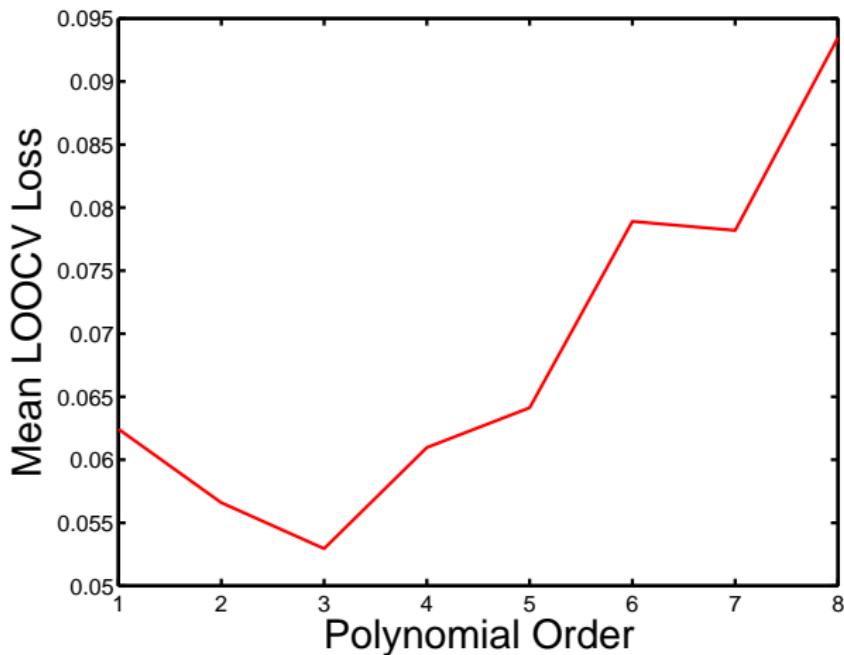
# Cross-validation



## Leave-one-out Cross-validation

- ▶ Extreme example: choose  $M = N$ 
  - ▶ each fold includes one input-response pair
  - ▶ **leave-one-out** (LOO) CV

## LOOCV – Olympic data



Best model?

LOO CV suggests a 3rd order model. Previous method suggests 1st order. Who knows which is right!

## Computational issues

- ▶ for  $M$ -fold CV, need to train our model  $M$  times
- ▶ for LOO-CV, need to train out model  $N$  times
- ▶ **computationally expensive!**

## Computational issues

- ▶ for  $M$ -fold CV, need to train our model  $M$  times
- ▶ for LOO-CV, need to train out model  $N$  times
- ▶ **computationally expensive!**
  
- ▶ for  $t = \mathbf{w}^T \mathbf{x}$ , this is feasible if  $K$  (number of terms in function) isn't too big:

$$\begin{aligned} t &= \sum_{k=0}^K w_k h_k(x) \\ \hat{\mathbf{w}} &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t} \end{aligned}$$

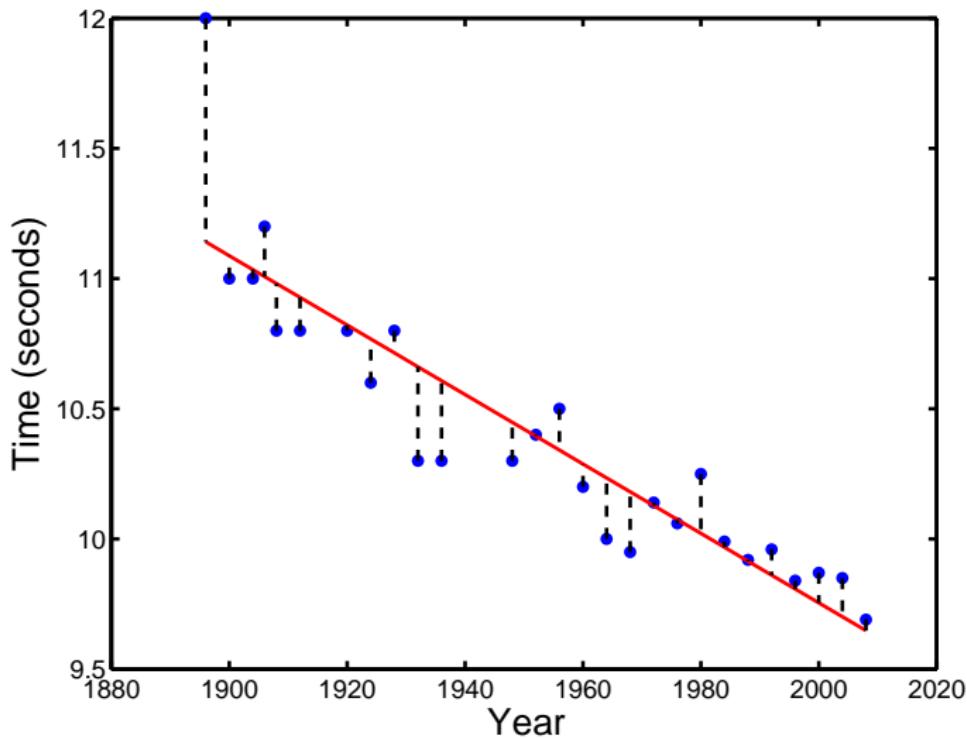
- ▶ for some models, we will need to use  $M \ll N$

## Summary

- ▶ Saw how choice of model has big influence in quality of predictions
- ▶ Saw how the loss on the training data,  $\mathcal{L}$ , cannot be used to choose models
  - ▶ Making model more complex always decreases the loss
- ▶ Introduced the idea of using some data for validation
- ▶ Introduced cross validation and leave-one-out cross validation

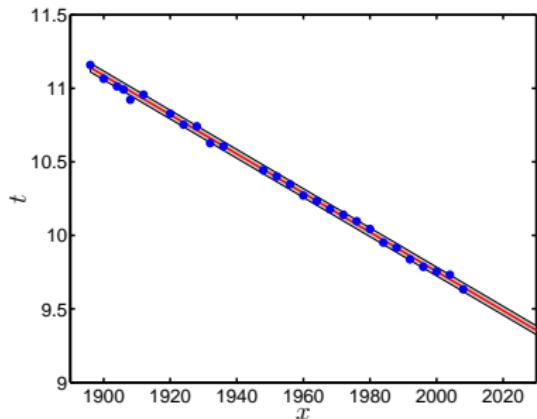
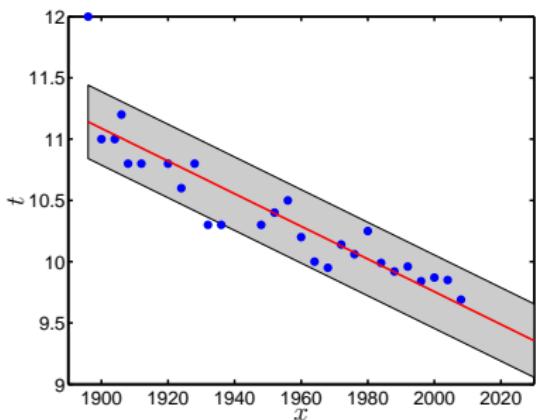
## What about the errors?

$$t = w_0 + w_1 x = \mathbf{w}^T \mathbf{x}$$



# We should model the errors!

- ▶ errors tell us how **confident** our predictions should be:



## We will...

- ▶ briefly recap random variables
- ▶ change our model to output a random variable
- ▶ introduce *likelihood* as a replacement for loss
- ▶ find the parameters that maximise the likelihood...
  - ▶ ...instead of minimising the loss

## $t_n$ as a *random variable*

- ▶ the actual winning time is still uncertain, even given our model
- ▶ there is some *error* for each year
  - ▶ ...due to inherent unpredictability in winning time
  - ▶ ...due to imperfect measurement
- ▶ instead: treat  $t_n$  as a *random variable*
- ▶ random variable = mathematical representation of a quantity that's uncertain

## Random variables

- ▶ Suppose I toss a coin and assign the variable  $X$  the value 1 if the coin lands heads and 0 if it lands tails
- ▶ ...then  $X$  is a **random variable**

## Random variables

- ▶ Suppose I toss a coin and assign the variable  $X$  the value 1 if the coin lands heads and 0 if it lands tails
- ▶ ...then  $X$  is a **random variable**
- ▶ We don't know which value  $X$  will take, but we do know the possible values and how likely they are

# Random variables

- ▶ Suppose I toss a coin and assign the variable  $X$  the value 1 if the coin lands heads and 0 if it lands tails
- ▶ ...then  $X$  is a **random variable**
- ▶ We don't know which value  $X$  will take, but we do know the possible values and how likely they are

## Notation

- ▶ random variables given capital letters:  $X$ ,  $Y$
- ▶ lowercase letters used for values they can take:  $x$ ,  $y$

## Discrete and continuous RVs

- ▶ **Discrete** = random events with outcomes that we can count
  - ▶ coin toss
  - ▶ rolling a die
  - ▶ next word in a document
  - ▶ number of emails sent in a day

# Discrete and continuous RVs

- ▶ **Discrete** = random events with outcomes that we can count
  - ▶ coin toss
  - ▶ rolling a die
  - ▶ next word in a document
  - ▶ number of emails sent in a day
- ▶ **Continuous** = random events with outcomes that we cannot count:
  - ▶ winning time in Olympic 100m
  - ▶ noise in our model!

## Discrete RVs

Discrete RVs defined by probabilities of different events taking place.  
e.g. probability of random variable  $X$  taking value  $x$ :

$$P(X = x)$$

For example, fair coin:

$$P(X = 1) = 0.5, \quad P(X = 0) = 0.5$$

Die:

$$P(Y = y) = \frac{1}{6} \quad \text{for } y = 1, \dots, 6$$

## Discrete RVs

Discrete RVs defined by probabilities of different events taking place.  
e.g. probability of random variable  $X$  taking value  $x$ :

$$P(X = x)$$

For example, fair coin:

$$P(X = 1) = 0.5, \quad P(X = 0) = 0.5$$

Die:

$$P(Y = y) = \frac{1}{6} \quad \text{for } y = 1, \dots, 6$$

Probabilities are constrained:

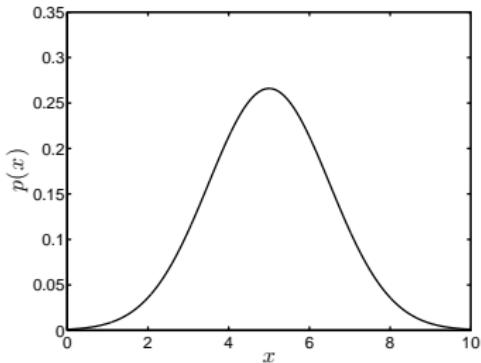
$$0 \leq P(Y = y) \leq 1, \quad \sum_y P(Y = y) = 1$$

## Continuous RVs

- ▶ Can't list all possible outcomes and probabilities!

# Continuous RVs

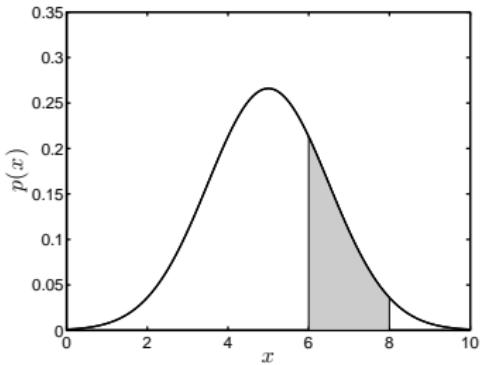
- ▶ Can't list all possible outcomes and probabilities!
- ▶ Instead define a density function  $p(x)$ :



- ▶  $p(x)$  tells us how likely different values are
- ▶ these are **not** probabilities!

# Continuous RVs

- ▶ Can't list all possible outcomes and probabilities!
- ▶ Instead define a density function  $p(x)$ :



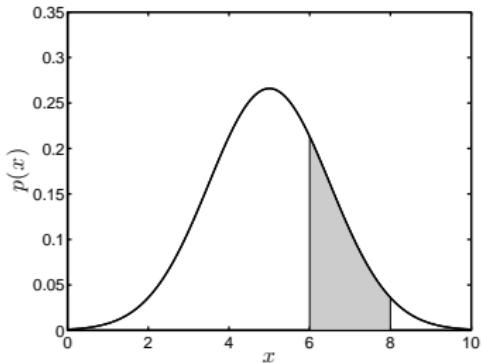
- ▶  $p(x)$  tells us how likely different values are
- ▶ these are **not** probabilities!

- ▶ We can compute probabilities of ranges by computing the area under the curve:

$$P(6 \leq X \leq 8) = \int_{x=6}^{x=8} p(x) \, dx$$

# Continuous RVs

- ▶ Can't list all possible outcomes and probabilities!
- ▶ Instead define a density function  $p(x)$ :



- ▶  $p(x)$  tells us how likely different values are
- ▶ these are **not** probabilities!

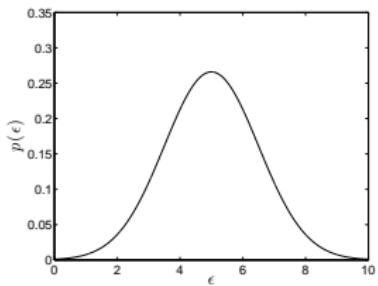
- ▶ We can compute probabilities of ranges by computing the area under the curve:

$$P(6 \leq X \leq 8) = \int_{x=6}^{x=8} p(x) \, dx$$

- ▶ Densities are constrained:

$$p(x) \geq 0, \quad \int_{-\infty}^{\infty} p(x) \, dx = 1$$

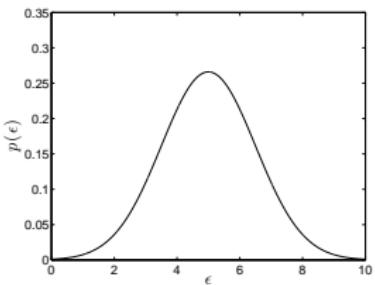
## Example: Gaussian RVs



$$\epsilon \sim \mathcal{N}(\mu, \sigma^2)$$

$$\begin{aligned} p(\epsilon|\mu, \sigma^2) &= f_{\mathcal{N}}(\mu, \sigma^2) \\ &= \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ -\frac{1}{2\sigma^2}(\epsilon - \mu)^2 \right\} \end{aligned}$$

## Example: Gaussian RVs



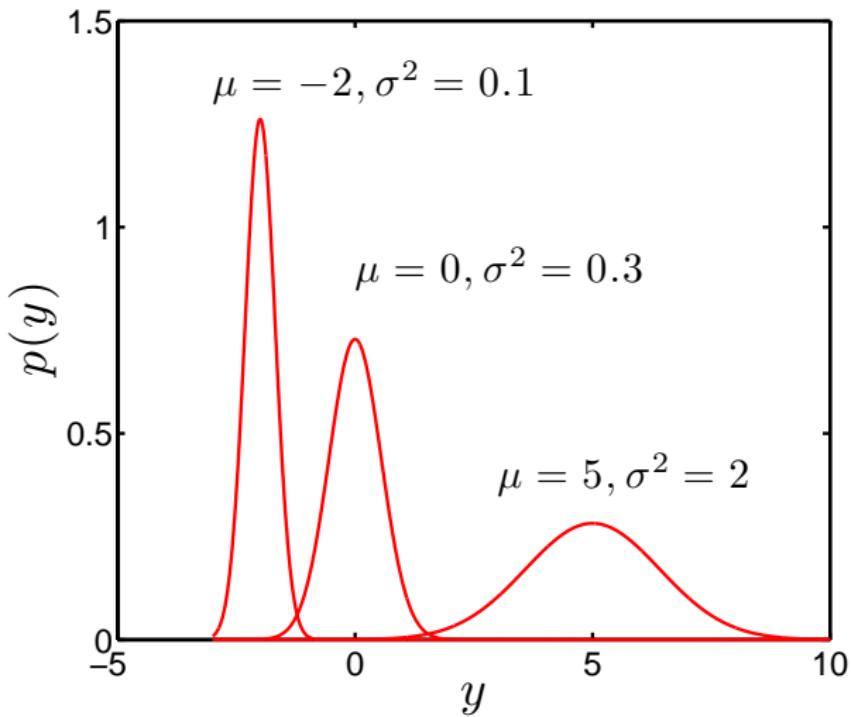
$$\epsilon \sim \mathcal{N}(\mu, \sigma^2)$$

$$p(\epsilon|\mu, \sigma^2) = f_{\mathcal{N}}(\mu, \sigma^2)$$

$$= \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ -\frac{1}{2\sigma^2}(\epsilon - \mu)^2 \right\}$$

- ▶ two parameters:  $\mu$  = **mean**, and  $\sigma^2$  = **variance**
- ▶  $\mu$  says where the peak is
- ▶  $\sigma^2$  says how wide it is

## Example: Gaussian RVs



Effect of varying the mean ( $\mu$ ) and variance ( $\sigma^2$ )

# Joint probabilities and densities

## Joint probabilities

For two discrete RVs,  $X$  and  $Y$ ,  $P(X = x, Y = y)$  is the probability that RV  $X$  has value  $x$  **and** RV  $Y$  has value  $y$ .

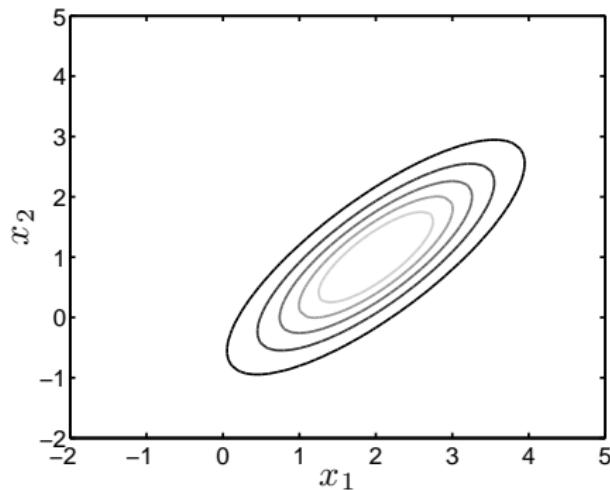
# Joint probabilities and densities

## Joint probabilities

For two discrete RVs,  $X$  and  $Y$ ,  $P(X = x, Y = y)$  is the probability that RV  $X$  has value  $x$  **and** RV  $Y$  has value  $y$ .

## Joint densities

For two continuous RVs,  $x_0$  and  $x_1$ ,  $p(x_0, x_1)$  is the joint density:



# Dependence & Independence

- ▶ Let  $X$  be the random variable for the toss of a coin
  - ▶ 1 = heads, 0 = tails
- ▶ Let  $Y$  be the random variable for the rolling of a die
- ▶  $P(X = 1, Y = 3)$  is the probability that I will roll a head **and** a 3

# Dependence & Independence

- ▶ Let  $X$  be the random variable for the toss of a coin
  - ▶ 1 = heads, 0 = tails
- ▶ Let  $Y$  be the random variable for the rolling of a die
- ▶  $P(X = 1, Y = 3)$  is the probability that I will roll a head **and** a 3
- ▶ The outcome of  $X$  does not depend on  $Y$

# Dependence & Independence

- ▶ Let  $X$  be the random variable for the toss of a coin
  - ▶ 1 = heads, 0 = tails
- ▶ Let  $Y$  be the random variable for the rolling of a die
- ▶  $P(X = 1, Y = 3)$  is the probability that I will roll a head **and** a 3
- ▶ The outcome of  $X$  does not depend on  $Y$
- ▶  $X$  and  $Y$  are **independent**

$$P(X = x, Y = y) = P(X = x)P(Y = y)$$

## Dependence & Independence

- ▶ Let  $X$  be the random variable for the event – I'm playing tennis
  - ▶ 1 = yes, 0 = no
- ▶ Let  $Y$  be the random variable for the event – It is raining
  - ▶ 1 = yes, 0 = no
- ▶  $P(X = 1, Y = 1)$  is the probability that I am playing and it is raining

## Dependence & Independence

- ▶ Let  $X$  be the random variable for the event – I'm playing tennis
  - ▶ 1 = yes, 0 = no
- ▶ Let  $Y$  be the random variable for the event – It is raining
  - ▶ 1 = yes, 0 = no
- ▶  $P(X = 1, Y = 1)$  is the probability that I am playing and it is raining
- ▶ The outcome of  $X$  **does** depend on  $Y$

# Dependence & Independence

- ▶ Let  $X$  be the random variable for the event – I'm playing tennis
  - ▶ 1 = yes, 0 = no
- ▶ Let  $Y$  be the random variable for the event – It is raining
  - ▶ 1 = yes, 0 = no
- ▶  $P(X = 1, Y = 1)$  is the probability that I am playing and it is raining
- ▶ The outcome of  $X$  **does** depend on  $Y$
- ▶  $X$  and  $Y$  are **dependent**

$$P(X = x, Y = y) \neq P(X = x)P(Y = y)$$

# Conditioning

- ▶ Let  $X$  be the random variable for the event – I'm playing tennis
  - ▶ 1 = yes, 0 = no
- ▶ Let  $Y$  be the random variable for the event – It is raining
  - ▶ 1 = yes, 0 = no
- ▶ We can look at **conditional** probabilities

# Conditioning

- ▶ Let  $X$  be the random variable for the event – I'm playing tennis
  - ▶ 1 = yes, 0 = no
- ▶ Let  $Y$  be the random variable for the event – It is raining
  - ▶ 1 = yes, 0 = no
- ▶ We can look at **conditional** probabilities
- ▶ e.g. the probability that I am playing **given that** it is raining:

$$P(X = 1 \mid Y = 1)$$

# Conditioning

- ▶ Let  $X$  be the random variable for the event – I'm playing tennis
  - ▶ 1 = yes, 0 = no
- ▶ Let  $Y$  be the random variable for the event – It is raining
  - ▶ 1 = yes, 0 = no
- ▶ We can look at **conditional** probabilities
- ▶ e.g. the probability that I am playing **given that** it is raining:

$$P(X = 1 \mid Y = 1)$$

- ▶ We can decompose the joint probability:

$$P(X = x, Y = y) = P(X = x \mid Y = y) P(Y = y)$$

## Back to our model...

- ▶ **before:** model predicted single value  $t_n = \mathbf{w}^T \mathbf{x}_n$
- ▶ **now:** model predicts random variable  $T_n$

## Back to our model...

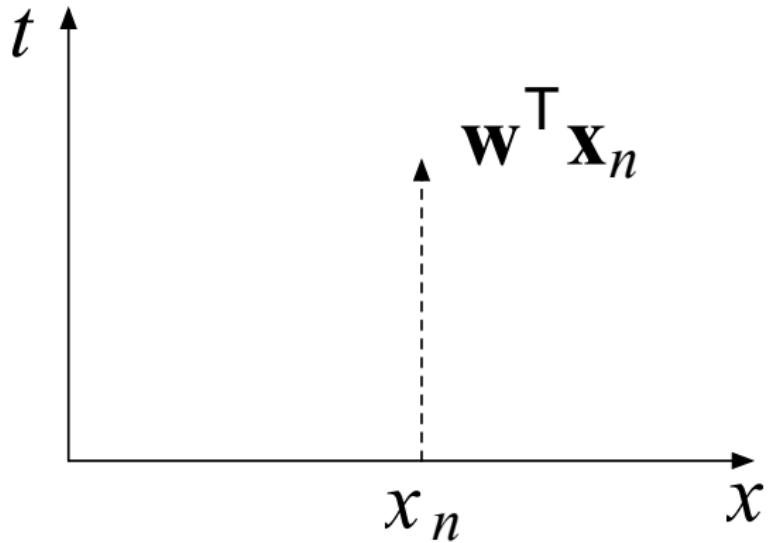
- ▶ **before:** model predicted single value  $t_n = \mathbf{w}^T \mathbf{x}_n$
- ▶ **now:** model predicts random variable  $T_n$

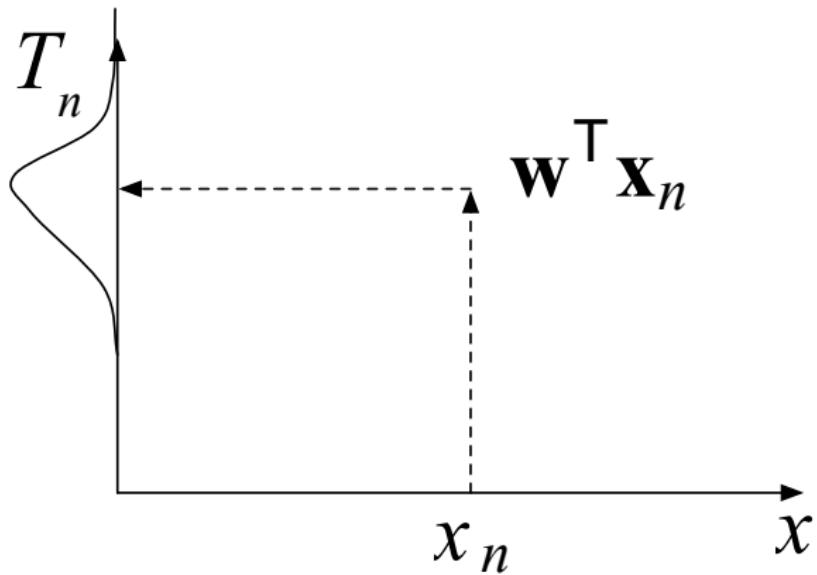
$$T_n = \mathbf{w}^T \mathbf{x}_n + \epsilon_n$$

- ▶  $\epsilon_n$  is the **noise**,  $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$

- ▶ equivalently:

$$T_n \sim \mathcal{N}(\mathbf{w}^T \mathbf{x}_n, \sigma^2)$$





- ▶  $T_n \sim \mathcal{N}(w^T x_n, \sigma^2)$
- ▶  $T_n = w^T x_n + \epsilon_n$ , where  $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$

## Likelihood

- ▶  $T_n$  is a Gaussian random variable

$$T_n \sim \mathcal{N}(\mathbf{w}^\top \mathbf{x}_n, \sigma^2)$$

- ▶ it has probability density

$$p(T_n = t \mid \mathbf{w}, \mathbf{x}_n, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(t - \mathbf{w}^\top \mathbf{x}_n)^2\right\}$$

# Likelihood

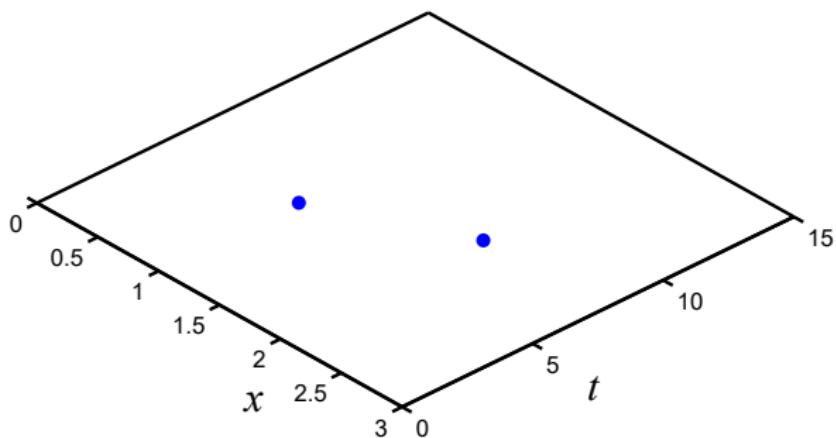
- ▶  $T_n$  is a Gaussian random variable

$$T_n \sim \mathcal{N}(\mathbf{w}^\top \mathbf{x}_n, \sigma^2)$$

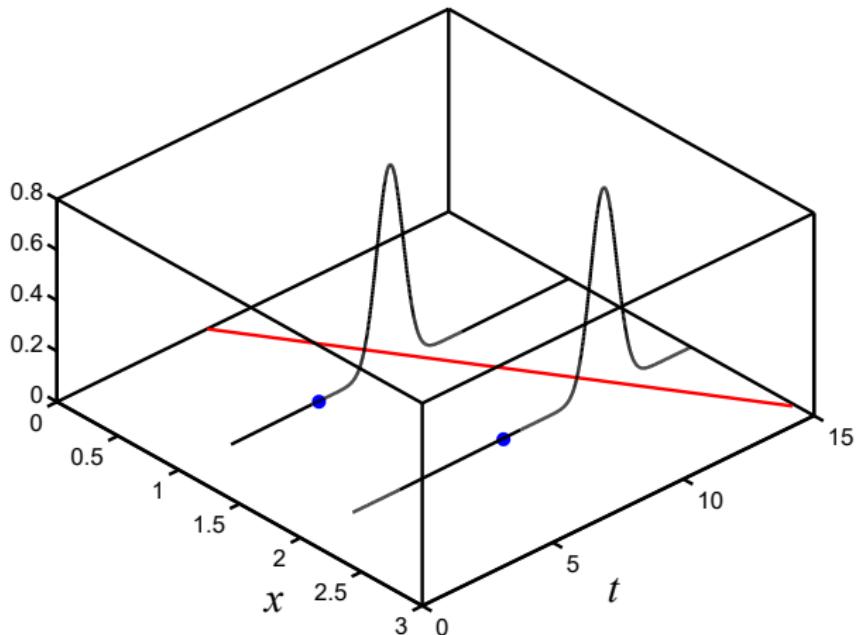
- ▶ it has probability density

$$p(T_n = t \mid \mathbf{w}, \mathbf{x}_n, \sigma^2) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left\{ -\frac{1}{2\sigma^2} (t - \mathbf{w}^\top \mathbf{x}_n)^2 \right\}$$

- ▶  $t_n$  is our (non-random!) observation
- ▶ density of  $T_n$  at point  $t_n$  is called **likelihood** of  $t_n$ 
  - ▶ i.e.  $p(T_n = t_n \mid \mathbf{w}, \mathbf{x}_n, \sigma^2)$
- ▶ vary  $\mathbf{w}$  to maximise the likelihood of  $t_n$

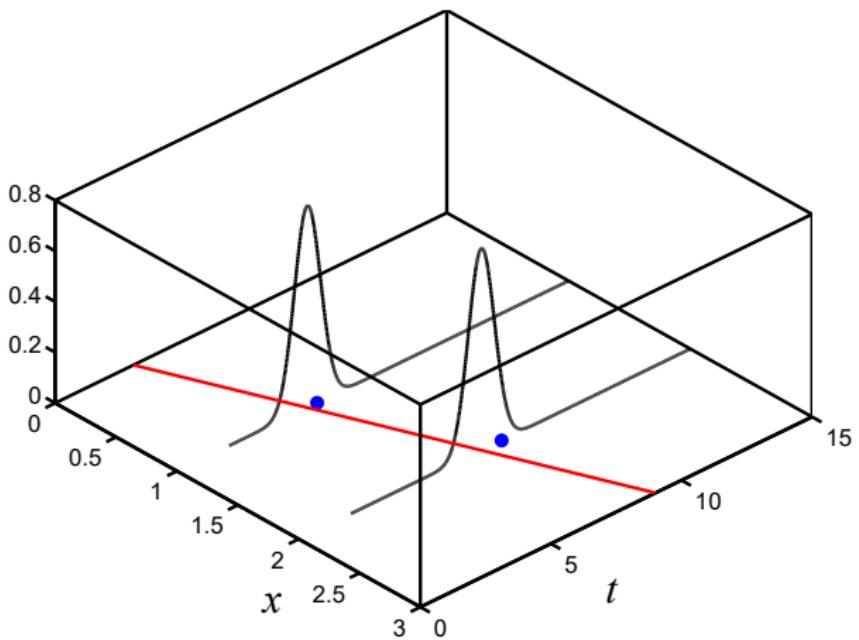


$$p(T_n = t_n \mid \mathbf{w}, \mathbf{x}_n, \sigma^2)$$



**Model 1: low likelihood**

$$p(T_n = t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2)$$



**Model 2: high likelihood**

## Likelihood optimisation

- ▶ For each input-response pair, we have a Gaussian likelihood

$$p(T_n = t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2)$$

- ▶ To combine them all, we want the joint likelihood:

$$p(T_1 = t_1, \dots, T_N = t_N | \mathbf{w}, \sigma^2, \mathbf{x}_1, \dots, \mathbf{x}_N)$$

# Likelihood optimisation

- ▶ For each input-response pair, we have a Gaussian likelihood

$$p(T_n = t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2)$$

- ▶ To combine them all, we want the joint likelihood:

$$p(T_1 = t_1, \dots, T_N = t_N | \mathbf{w}, \sigma^2, \mathbf{x}_1, \dots, \mathbf{x}_N)$$

- ▶ Assume that the  $t_n$  are independent:

$$p(T_1 = t_1, \dots, T_N = t_N | \mathbf{w}, \sigma^2, \mathbf{x}_1, \dots, \mathbf{x}_N) = \prod_{n=1}^N p(T_n = t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2)$$

# Likelihood optimisation

Find the parameters that maximise the joint likelihood:

$$\operatorname{argmax}_{\mathbf{w}, \sigma^2} \prod_{n=1}^N p(T_n = t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2)$$

# Likelihood optimisation

Find the parameters that maximise the joint likelihood:

$$\operatorname{argmax}_{\mathbf{w}, \sigma^2} \prod_{n=1}^N p(T_n = t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2)$$

Easier: optimise the log-likelihood:

- ▶ if we increase  $z$ ,  $\log(z)$  increases
- ▶ if we decrease  $z$ ,  $\log(z)$  decreases
- ▶ so, at a maximum of  $z$ ,  $\log(z)$  will also be at a maximum

# Likelihood optimisation

Find the parameters that maximise the joint likelihood:

$$\operatorname{argmax}_{\mathbf{w}, \sigma^2} \prod_{n=1}^N p(T_n = t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2)$$

Easier: optimise the log-likelihood:

- ▶ if we increase  $z$ ,  $\log(z)$  increases
- ▶ if we decrease  $z$ ,  $\log(z)$  decreases
- ▶ so, at a maximum of  $z$ ,  $\log(z)$  will also be at a maximum

$$\operatorname{argmax}_{\mathbf{w}, \sigma^2} \log \prod_{n=1}^N p(T_n = t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2)$$

## Some re-arranging...

$$\begin{aligned} p(T_n = t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2) &= \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ -\frac{1}{2\sigma^2} (t_n - \mathbf{w}^\top \mathbf{x}_n)^2 \right\} \\ \log L &= \log \prod_{n=1}^N p(t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2) \end{aligned}$$

## Some re-arranging...

$$\begin{aligned} p(T_n = t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2) &= \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ -\frac{1}{2\sigma^2} (t_n - \mathbf{w}^\top \mathbf{x}_n)^2 \right\} \\ \log L &= \log \prod_{n=1}^N p(t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2) \\ &= \sum_{n=1}^N \log p(t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2) \\ &= \sum_{n=1}^N \log \left( \frac{1}{\sigma\sqrt{2\pi}} \right) - \sum_{n=1}^N \frac{1}{2\sigma^2} (t_n - \mathbf{w}^\top \mathbf{x}_n)^2 \\ &= -N \log(\sigma\sqrt{2\pi}) - \frac{1}{2\sigma^2} \sum_{n=1}^N (t_n - \mathbf{w}^\top \mathbf{x}_n)^2 \end{aligned}$$

## Some re-arranging...

$$\begin{aligned} p(T_n = t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2) &= \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ -\frac{1}{2\sigma^2} (t_n - \mathbf{w}^\top \mathbf{x}_n)^2 \right\} \\ \log L &= \log \prod_{n=1}^N p(t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2) \\ &= -N \log(\sigma\sqrt{2\pi}) - \frac{1}{2\sigma^2} \sum_{n=1}^N (t_n - \mathbf{w}^\top \mathbf{x}_n)^2 \end{aligned}$$

## Some re-arranging...

$$\begin{aligned} p(T_n = t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2) &= \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(t_n - \mathbf{w}^\top \mathbf{x}_n)^2\right\} \\ \log L &= \log \prod_{n=1}^N p(t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2) \\ &= -N \log(\sigma\sqrt{2\pi}) - \frac{1}{2\sigma^2} \sum_{n=1}^N (t_n - \mathbf{w}^\top \mathbf{x}_n)^2 \end{aligned}$$

- ▶ Looks familiar! To continue (good exercise):

$$\frac{\partial \log L}{\partial \mathbf{w}} = 0, \quad \frac{\partial \log L}{\partial \sigma^2} = 0$$

## Optimum parameters

- ▶ Compute optimum  $\hat{\mathbf{w}}$  from:

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}$$

## Optimum parameters

- ▶ Compute optimum  $\hat{\mathbf{w}}$  from:

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}$$

- ▶ Use this to compute optimum  $\hat{\sigma}^2$  from:

$$\hat{\sigma}^2 = \frac{1}{N} (\mathbf{t} - \mathbf{X}\hat{\mathbf{w}})^T (\mathbf{t} - \mathbf{X}\hat{\mathbf{w}})$$

# Optimum parameters

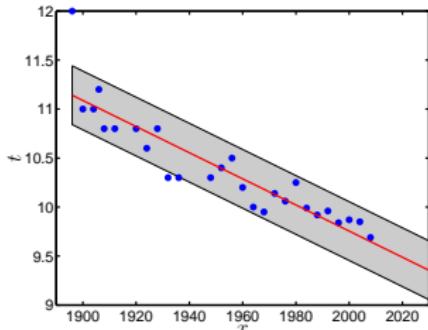
- ▶ Compute optimum  $\hat{\mathbf{w}}$  from:

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}$$

- ▶ Use this to compute optimum  $\hat{\sigma}^2$  from:

$$\hat{\sigma}^2 = \frac{1}{N} (\mathbf{t} - \mathbf{X}\hat{\mathbf{w}})^T (\mathbf{t} - \mathbf{X}\hat{\mathbf{w}})$$

- ▶ e.g. Olympic 100m data (again!)



$$\hat{\mathbf{w}} = \begin{bmatrix} 36.416 \\ -0.0133 \end{bmatrix}, \hat{\sigma}^2 = 0.0503$$

## Summary

- ▶ The quantity obtained when evaluating the density  $p(T_n = t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2)$  is called the **likelihood**

## Summary

- ▶ The quantity obtained when evaluating the density  $p(T_n = t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2)$  is called the **likelihood**
- ▶ The higher the value, the more likely  $t_n$  is given the model...
  - ▶ ...the better the model is

## Summary

- ▶ The quantity obtained when evaluating the density  $p(T_n = t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2)$  is called the **likelihood**
- ▶ The higher the value, the more likely  $t_n$  is given the model...
  - ▶ ...the better the model is
- ▶ Remember: It is **not** a probability!

## Summary

- ▶ The quantity obtained when evaluating the density  $p(T_n = t_n | \mathbf{w}, \mathbf{x}_n, \sigma^2)$  is called the **likelihood**
- ▶ The higher the value, the more likely  $t_n$  is given the model...
  - ▶ ...the better the model is
- ▶ Remember: It is **not** a probability!
- ▶ For fixed  $t_n$  and  $x_n$ , we can find the values of  $\mathbf{w}$  and  $\sigma^2$  that maximise the likelihood
  - ▶ ...just like previously we minimised the loss