Uncertainty in Deep Learning

Yarin Gal

University of Oxford
yarin@cs.ox.ac.uk

Relevant resources: yr.gl/udl23

SLIDES

Slide decks from the talks.

SLIDE DECK 1

SLIDE DECK 2

DEMO

Demoes mentioned in the slides

UNCERTAINTY PLAYGROUND

UNCERTAINTY VISUALISATION

RECAP

A quick recap of useful stuff.

GAUSSIANS RECAP

NOTATION

Notation used in the slides:.

MORE STUFF

OATML

Bayesian Probabilistic Modelling (an Introduction)

Simple idea: *"If you're doing something which doesn't follow from the laws of probability, then you're doing it wrong"*

Structure

- From beliefs to ML
- On ML and ‘assumptions’
- Generative story and probabilistic model
- Intuition: what does the likelihood really mean? (likelihood as a function of parameters)
- Intuition: The Posterior

From beliefs to ML

- I measured the temp this morning with my phone, phone said 8c
- My phone's temp sensor is noisy though.. (spec says it has 5c standard deviation)
- What do you think the actual temp was? (I don't want a single number!)

From beliefs to ML

- We can write this down in maths
  - "I measured temp $x = 8$; sensor is noisy $\sigma = 5$; True temp μ?"
    We call this the likelihood [how I believe data/obs was generated given params]

$$X \mid \mu, \sigma = 5 \sim \mathcal{N}(\mu, \sigma^2).$$

- Since rational beliefs must follow the laws of prob theory, given our observation $x = 8$ we can encode our belief about what the true temp μ might have been using *Bayes law* **[whiteboard]**

$$\overbrace{p(\mu|\mathcal{D} = \{8\}, \sigma = 5)}^{\text{Posterior}} = \frac{\overbrace{p(\mathcal{D} = \{8\}|\mu, \sigma = 5)}^{\text{Likelihood}}\overbrace{p(\mu|\sigma = 5)}^{\text{Prior}}}{\underbrace{p(\mathcal{D} = \{8\}|\sigma = 5)}_{\text{Model evidence}}}$$

- The prior above is what I thought the temp might be before making my obs, eg $\mu \sim \mathcal{N}(5, 10)$. This will make more sense next

UNIVERSITY OF OXFORD

- We can write this down in maths
  - "I measured temp $x = 8$; sensor is noisy $\sigma = 5$; True temp μ?" We call this the likelihood [how I believe data/obs was generated given params]

$$X \mid \mu, \sigma = 5 \sim \mathcal{N}(\mu, \sigma^2).$$

- Since rational beliefs must follow the laws of prob theory, given our observation $x = 8$ we can encode our belief about what the true temp μ might have been using *Bayes law* **[whiteboard]**

$$\overbrace{p(\mu|\mathcal{D} = \{8\}, \sigma = 5)}^{\text{Posterior}} = \frac{\overbrace{p(\mathcal{D} = \{8\}|\mu, \sigma = 5)}^{\text{Likelihood}}\overbrace{p(\mu|\sigma = 5)}^{\text{Prior}}}{\underbrace{p(\mathcal{D} = \{8\}|\sigma = 5)}_{\text{Model evidence}}}$$

- The prior above is what I thought the temp might be before making my obs, eg $\mu \sim \mathcal{N}(5, 10)$. This will make more sense next

From beliefs to ML

- We can write this down in maths
  - "I measured temp $x = 8$; sensor is noisy $\sigma = 5$; True temp μ?" We call this the likelihood [how I believe data/obs was generated given params]

$$X \mid \mu, \sigma = 5 \sim \mathcal{N}(\mu, \sigma^2).$$

- Since rational beliefs must follow the laws of prob theory, given our observation $x = 8$ we can encode our belief about what the true temp μ might have been using *Bayes law* **[whiteboard]**

$$\overbrace{p(\mu|\mathcal{D} = \{8\}, \sigma = 5)}^{\text{Posterior}} = \frac{\overbrace{p(\mathcal{D} = \{8\}|\mu, \sigma = 5)}^{\text{Likelihood}}\overbrace{p(\mu|\sigma = 5)}^{\text{Prior}}}{\underbrace{p(\mathcal{D} = \{8\}|\sigma = 5)}_{\text{Model evidence}}}$$

- The prior above is what I thought the temp might be before making my obs, eg $\mu \sim \mathcal{N}(5, 10)$. This will make more sense next

From beliefs to ML

- I made a second measurement with my phone, and this time it said 13c
- Remember: phone's temp sensor is noisy.. (5c standard deviation)
- What do you think the actual temp μ was now?

From beliefs to ML

- Our posterior belief from before is our new *prior belief*

$$\overbrace{p(\mu|\mathcal{D} = \{8\}, \sigma = 5)}^{\text{Prior}}$$

- Which, given the new obs $x = 13$, we must update using the laws of probability:

$$\overbrace{p(\mu|\mathcal{D} = \{8, 13\}, \sigma = 5)}^{\text{Posterior}} = \frac{\overbrace{p(\mathcal{D} = \{8, 13\}|\mu, \sigma = 5)}^{\text{Likelihood}}\overbrace{p(\mu|\mathcal{D} = \{8\}, \sigma =}^{\text{Prior}}}{\underbrace{p(\mathcal{D} = \{8, 13\}|\sigma = 5)}_{\text{Model evidence}}}$$

- And so on... (we'll see soon how to eval these things)

From beliefs to ML

- Our posterior belief from before is our new *prior belief*

$$\overbrace{p(\mu|\mathcal{D}=\{8\},\sigma=5)}^{\text{Prior}}$$

- Which, given the new obs $x = 13$, we must update using the laws of probability:

$$\overbrace{p(\mu|\mathcal{D}=\{8,13\},\sigma=5)}^{\text{Posterior}} = \frac{\overbrace{p(\mathcal{D}=\{8,13\}|\mu,\sigma=5)}^{\text{Likelihood}}\overbrace{p(\mu|\mathcal{D}=\{8\},\sigma=}^{\text{Prior}}}{\underbrace{p(\mathcal{D}=\{8,13\}|\sigma=5)}_{\text{Model evidence}}}$$

- And so on... (we'll see soon how to eval these things)

Probabilistic Modelling and Assumptions

Above is an example of encoding our assumptions into a *model*.

- can't do ML without **assumptions**
  - there always exists some **underlying process** that generated obs
  - **all models** make assumptions about this process, either explicitly, or implicitly
  - in Bayesian probabilistic modelling we make our assumptions about underlying process **explicit**
  - want to **infer** underlying process (find dist that generated data)

Probabilistic Modelling and Assumptions

Above is an example of encoding our assumptions into a *model*.

- can't do ML without **assumptions**
  - there always exists some **underlying process** that generated obs
  - **all models** make assumptions about this process, either explicitly, or implicitly
  - in Bayesian probabilistic modelling we make our assumptions about underlying process **explicit**
  - want to **infer** underlying process (find dist that generated data)

- eg – astrophysics: gravitational lensing
  - there exists a physics process magnifying far away galaxies
  - Nature chose lensing coeff $\rightarrow$ gravitational lensing mechanism $\rightarrow$ transform galaxy
  - We **observe** transformed galaxies, want to **infer** lensing ceoff

Probabilistic Modelling and Assumptions

Above is an example of encoding our assumptions into a *model*.

- can't do ML without **assumptions**
  - there always exists some **underlying process** that generated obs
  - **all models** make assumptions about this process, either explicitly, or implicitly
  - in Bayesian probabilistic modelling we make our assumptions about underlying process **explicit**
  - want to **infer** underlying process (find dist that generated data)

- eg – cats vs dogs classification
  - there exist some underlying rules we don't know
  - eg "if has pointy ears then cat"
  - We **observe** pairs (image, "cat"/"no cat"), and want to **infer** underlying mapping from images to labels

Probabilistic Modelling and Assumptions

- can't do ML without **assumptions**
  - there always exists some **underlying process** that generated obs
  - **all models** make assumptions about this process, either explicitly, or implicitly
  - in Bayesian probabilistic modelling we make our assumptions about underlying process **explicit**
  - want to **infer** underlying process (find dist that generated data)
- eg – Gaussian density estimation
  - I tell you the process I used to generate data and give 5 data points

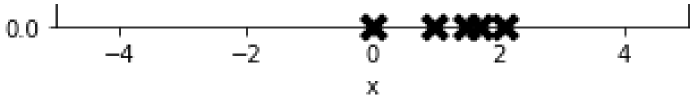
$$x_n \sim \mathcal{N}(x_n; \mu, \sigma^2), \qquad \sigma = 1$$

  - you **observe** the points $\{x_1, ..., x_5\}$, and want to **infer** my μ
  - Reminder: Gaussian density with mean μ and variance σ^2

$$p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Probabilistic Modelling and Assumptions

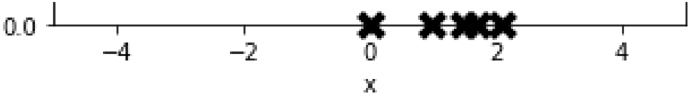
- can't do ML without **assumptions**
  - there always exists some **underlying process** that generated obs
  - **all models** make assumptions about this process, either explicitly, or implicitly
  - in Bayesian probabilistic modelling we make our assumptions about underlying process **explicit**
  - want to **infer** underlying process (find dist that generated data)
- eg – Gaussian density estimation



X Which μ generated my data?

V What's the probability that $\mu = 10$ generated my data? (want to infer distribution over μ!)

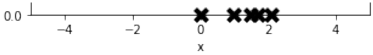
- can't do ML without **assumptions**
  - there always exists some **underlying process** that generated obs
  - **all models** make assumptions about this process, either explicitly, or implicitly
  - in Bayesian probabilistic modelling we make our assumptions about underlying process **explicit**
  - want to **infer** underlying process (find dist that generated data)
- eg – Gaussian density estimation



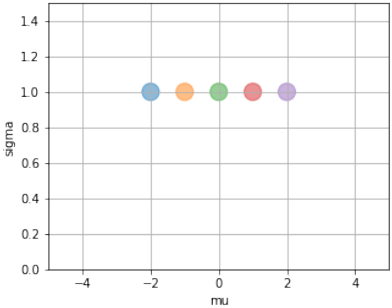
X Which μ generated my data?

V What's the probability that $\mu = 10$ generated my data? (want to infer distribution over μ!)

- My data:

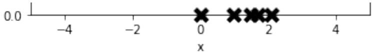


- These are the hypotheses we'll play with: (in parameter space)

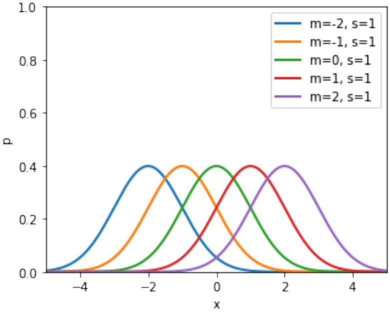


- I chose a Gaussian (one of those) from which I generated my data
- Which Gaussian do you think I chose?

- My data:

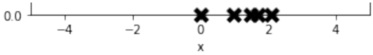


- And this is what the hypotheses look like in data space:

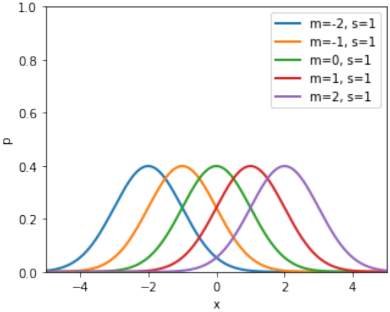


- I chose a Gaussian (one of those) from which I generated my data
- Which Gaussian do you think I chose?

- My data:



- And this is what the hypotheses look like in data space:



- I chose a Gaussian (one of those) from which I generated my data
- Which Gaussian do you think I chose?

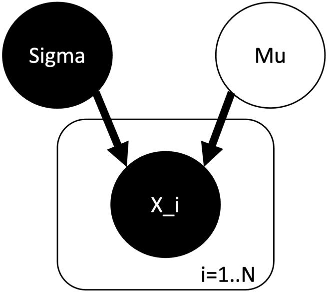
In Bayesian probabilistic modelling

- want to represent our beliefs / assumptions about how data was generated explicitly
- eg via *generative story* ['*My assumptions are...*']:
  - Someone (me / Nature / etc) selected parameters μ, σ
  - Generated N data points $x_n \sim \mathcal{N}(\mu, \sigma^2)$
  - Gave us $\mathcal{D} = \{x_1, ..., x_N\}$ (and also $\sigma = 1$)

In Bayesian probabilistic modelling

- want to represent our beliefs / assumptions about how data was generated explicitly
- eg via *generative story* ['*My assumptions are...*']:
  - Someone (me / Nature / etc) selected parameters μ, σ
  - Generated N data points $x_n \sim \mathcal{N}(\mu, \sigma^2)$
  - Gave us $\mathcal{D} = \{x_1, ..., x_N\}$ (and also $\sigma = 1$)

- Can represent generative story pictorially as a *plate diagram* (also known as a Bayesian network / graphical model)



  - Circle = random variable; plate = repetition;
  - Black = observed; white = unobserved;

- This represents how we believe our data was generated – what depends on what:
  - we're given σ, there exists *some* μ, and our observed x_i's are dependent on both

Joint distribution

Mathematically, this corresponds to this joint distribution:

$$p(\mathcal{D} = \{x_1, .., x_N\} | \mu, \sigma) \cdot p(\mu) \cdot p(\sigma)$$

with (product rule)

$$p(\mathcal{D} = \{x_1, .., x_N\} | \mu, \sigma) = \prod_{i=1}^{N} p(x_n | \mu, \sigma)$$

But we haven't specified what these distributions are yet... (e.g., what's $p(\mu)$?)

Generative story → probabilistic model

We can build a Bayesian probabilistic model following this generative story:

- prior [what I believe params might look like]

$$\mu \sim \mathcal{N}(0, 10), \qquad \sigma = 1$$

- likelihood [how I believe data was generated given params]

$$x_n \mid \mu, \sigma \sim \mathcal{N}(\mu, \sigma^2), \qquad n = 1..N$$

And plug these into the above joint distribution:

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

$$p(\mu) = \mathcal{N}(\mu; 0, 10)$$

and

$$p(\sigma) = \delta(\sigma = 1)$$

(σ equals 1 with prob 1)

Exercise: Draw a plate diagram, write the joint probability, and build a prob model for this generative story

My assumptions are...

- Someone (me / Nature / etc) selected parameters $\mu_n \in \mathbb{R}^{10}, \sigma_n \in \mathbb{R}^+$ for $n = 1..N$, and decoder function $f : \mathbb{R}^{10} \to X$
- Generated N latent points $z_n \sim \mathcal{N}(\mu_n, \sigma_n^2)$
- Decoded these z's and generated N data points $x_n \sim \mathcal{N}(f(z_n), 1)$
- Gave us $f, \mathcal{D} = \{x_1, ..., x_N\}$

Can you find my Gaussian?

How can we update prior belief on μ conditioned on data you give me?
(infer posterior distribution over $\mu \mid \{x_1, ..., x_N\}$)

Everything follows the **laws of prob**..

- Sum rule

$$p(X = x) = \sum_y p(X = x, Y = y) = \int p(X = x, Y) \mathrm{d}Y$$

- Product rule

$$p(X = x, Y = y) = p(X = x | Y = y) p(Y = y)$$

- Bayes rule

$$p(X = x | Y = y, \mathcal{H}) = \frac{p(Y = y | X = x, \mathcal{H}) p(X = x | \mathcal{H})}{p(Y = y | \mathcal{H})}$$

Note: $\mathcal{H}$ is often omitted in conditional for brevity

Remember: products, ratios, marginals, and conditionals of Gaussians are Gaussian!

Properties of Gaussian distributions:

If x_1, x_2 follow a joint Gaussian distribution:

$$\begin{bmatrix} x_1, \\ x_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1, \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12}^T & \Sigma_{22} \end{bmatrix} \right),$$

then each marginal is Gaussian:

$$x_1 \sim \mathcal{N}(\mu_1, \Sigma_{11}),$$

each conditional is Gaussian:

$$x_1 | x_2 \sim \mathcal{N}(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}\Sigma_{21}^T),$$

any linear combination is Gaussian:

$$Ax_1 + Bx_2 + C \sim \mathcal{N}(A\mu_1 + B\mu_2 + C, A\Sigma_{11}A^T + B\Sigma_{22}B^T)$$

and the product of the marginal densities is an (unnormalised) Gaussian:

$$\mathcal{N}(x; \mu_1, \Sigma_{11})\mathcal{N}(x; \mu_2, \Sigma_{22}) = C \cdot \mathcal{N}\left(x; (\Sigma_{11}^{-1} + \Sigma_{22}^{-1})^{-1}(\Sigma_{11}^{-1}\mu_1 + \Sigma_{22}^{-1}\mu_2), (\Sigma_{11}^{-1} + \Sigma_{22}^{-1})^{-1}\right)$$

with $C = \mathcal{N}(\mu_1; \mu_2, \Sigma_{11} + \Sigma_{22})$.

More here.

Visualising Gaussian likelihoods:

Summary (and playgroud) here: yr.gl/udl101

Updating our beliefs (Inference)

Bayes rule:

$$p(X = x | Y = y, \mathcal{H}) = \frac{p(Y = y | X = x, \mathcal{H}) p(X = x | \mathcal{H})}{p(Y = y | \mathcal{H})},$$

and in probabilistic modelling:

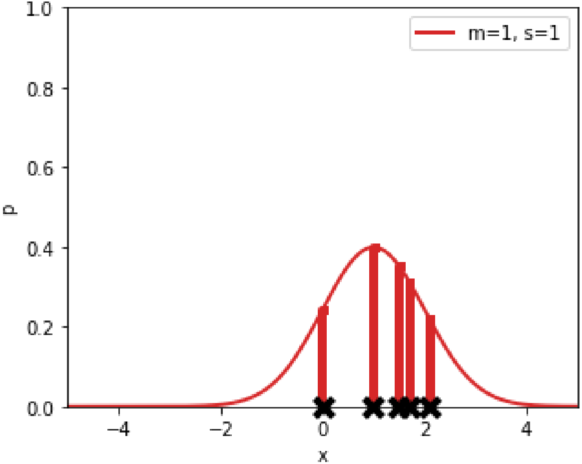
$$\overbrace{p(\mu | \mathcal{D}, \sigma, \mathcal{H})}^{\text{Posterior}} = \frac{\overbrace{p(\mathcal{D} | \mu, \sigma, \mathcal{H})}^{\text{Likelihood}} \overbrace{p(\mu | \sigma, \mathcal{H})}^{\text{Prior}}}{\underbrace{p(\mathcal{D} | \sigma, \mathcal{H})}_{\text{Model evidence}}}$$

with model evidence $p(\mathcal{D} | \sigma, \mathcal{H}) = \int p(\mathcal{D} | \mu, \sigma, \mathcal{H}) p(\mu | \sigma, \mathcal{H}) \mathrm{d}\mu$ (sum rule).

Let's look into each one of these quantities in more detail.

Likelihood $p(\mathcal{D}|\mu, \sigma, \mathcal{H})$

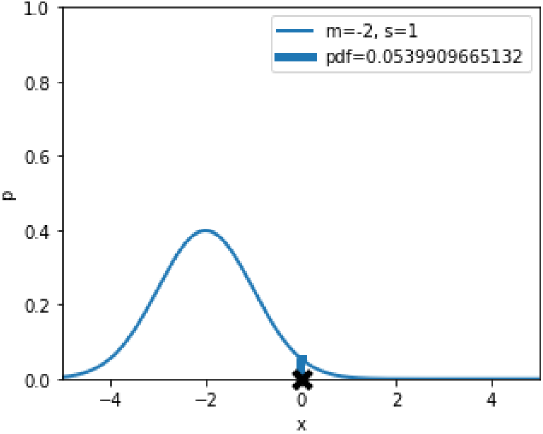
- we explicitly assumed data comes iid from a Gaussian
- compute $p(\mathcal{D}|\mu, \sigma)$ = multiply all $p(x_n|\mu, \sigma)$ (product rule)



Likelihood as a function of parameters

Let's look in detail – reducing dataset from 5 points to 1:

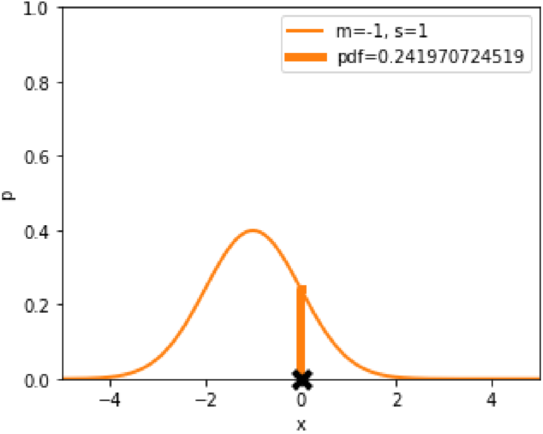
- What does the likelihood look like? let's 'scan' the parameter space



Likelihood as a function of parameters

Let's look in detail – reducing dataset from 5 points to 1:

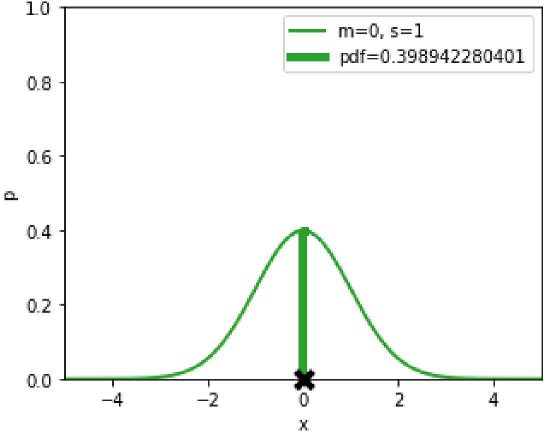
- What does the likelihood look like? let's 'scan' the parameter space



Likelihood as a function of parameters

Let's look in detail – reducing dataset from 5 points to 1:

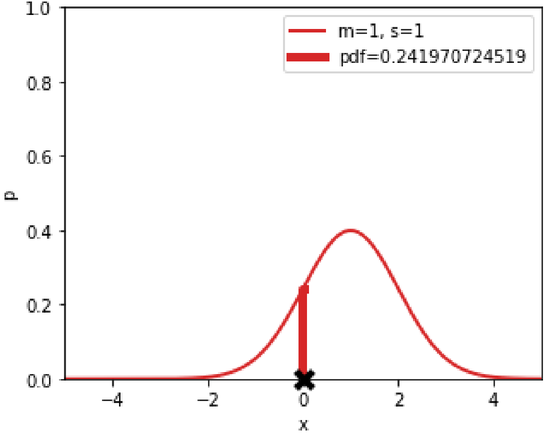
- What does the likelihood look like? let's 'scan' the parameter space



Likelihood as a function of parameters

Let's look in detail – reducing dataset from 5 points to 1:

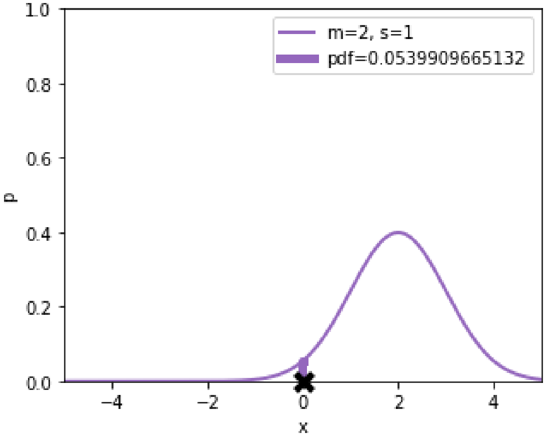
- What does the likelihood look like? let's 'scan' the parameter space



Likelihood as a function of parameters

Let's look in detail – reducing dataset from 5 points to 1:

- What does the likelihood look like? let's 'scan' the parameter space

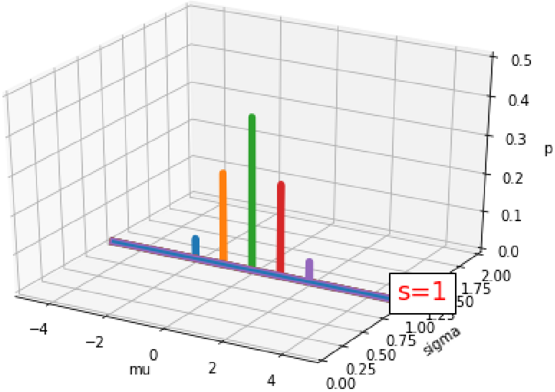


Likelihood as a function of parameters

Let's look in detail – reducing dataset from 5 points to 1:

- What does the likelihood look like? let's 'scan' the parameter space

  This is what the likelihood look like as a *function* of the params

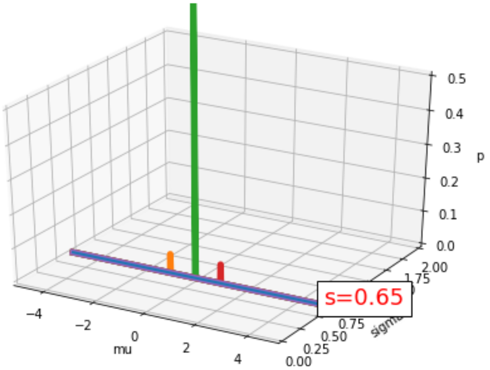


Likelihood as a function of parameters

Let's look in detail – reducing dataset from 5 points to 1:

- What does the likelihood look like? let's 'scan' the parameter space

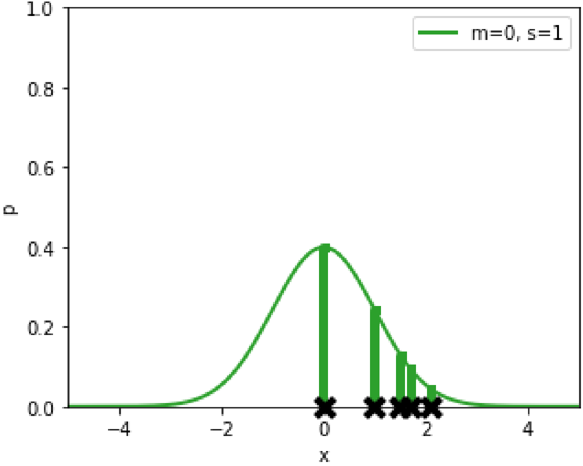
and with smaller σ..



- MLE (max lik) will get "absolutely certain that $\sigma = 0$ & $\mu = 0$"
- Does this make sense? (I **told** you $x_n \sim \mathcal{N}$!)
- MLE failure

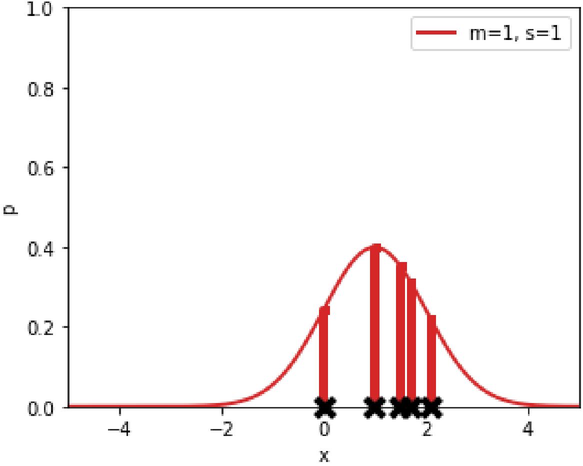
Likelihood as a function of parameters

And now scanning with all data, computing product of likelihoods:



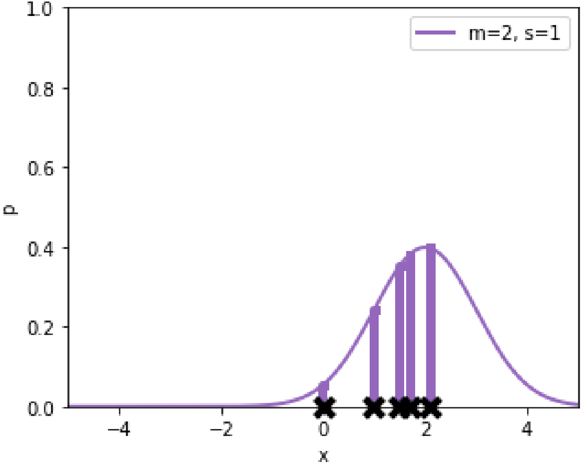
Likelihood as a function of parameters

And now scanning with all data, computing product of likelihoods:

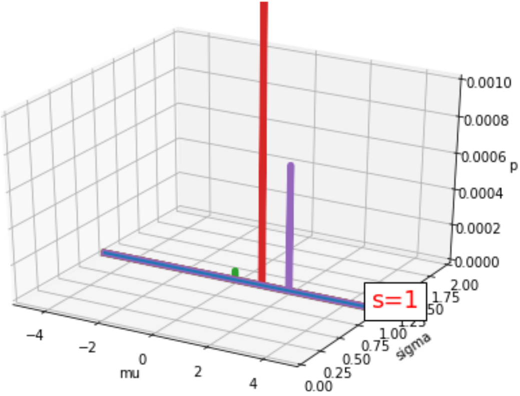


Likelihood as a function of parameters

And now scanning with all data, computing product of likelihoods:



And now scanning with all data, computing product of likelihoods:



Likelihood function tells us how well every value of μ, σ predicted what would happen.

The Posterior in more detail

$$\overbrace{p(\mu|\mathcal{D},\sigma,\mathcal{H})}^{\text{Posterior}} = \frac{\overbrace{p(\mathcal{D}|\mu,\sigma,\mathcal{H})}^{\text{Likelihood}}\overbrace{p(\mu|\sigma,\mathcal{H})}^{\text{Prior}}}{\underbrace{p(\mathcal{D}|\sigma,\mathcal{H})}_{\text{Model evidence}}}$$

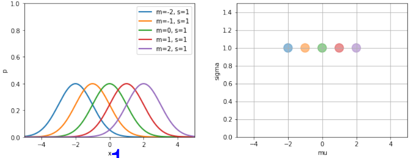
In contrast to the likelihood, posterior would say

'*with the data you gave me and prior assumptions, this is what I think μ could be, and I might become more confident if you give me more data*'

- normaliser = **marginal** likelihood = evidence = sum of likelihood * prior
  - $p(\mathcal{D}|\sigma,\mathcal{H}) = \int p(\mathcal{D}|\mu,\sigma,\mathcal{H})p(\mu|\sigma,\mathcal{H})d\mu$ (sum rule)
  - (but often difficult to calculate... more in the next lecture)

UNIVERSITY OF OXFORD

- Eg, inference w prior = 'we believe data is equally likely to have come from one of the 5 Gaussians w $\sigma = 1$'



$$p(\mu = \mu_i|\sigma, \mathcal{H}) = \frac{1}{5} \text{ and } p(\mu \neq \mu_i \text{ for all } i|\sigma, \mathcal{H}) = 0$$

then marginal likelihood is

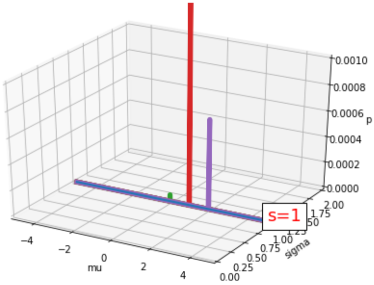
$$p(\mathcal{D}|\sigma, \mathcal{H}) = \sum_i p(\mathcal{D}|\mu = \mu_i, \sigma, \mathcal{H}) p(\mu = \mu_i|\sigma, \mathcal{H})$$

$$= \sum_i p(\mathcal{D}|\mu = \mu_i, \sigma, \mathcal{H}) \frac{1}{5}$$

and posterior is

$$p(\mu = \mu_i|\sigma, \mathcal{D}, \mathcal{H}) = \frac{1/5 p(\mathcal{D}|\mu = \mu_i, \sigma, \mathcal{H})}{\sum_i 1/5 p(\mathcal{D}|\mu = \mu_i, \sigma, \mathcal{H})}$$

Using likelihood from above, our posterior distribution over μ (prob of different values for μ given the observed data) is given by:



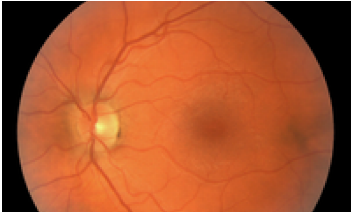
Some useful terminology we will use in the future:

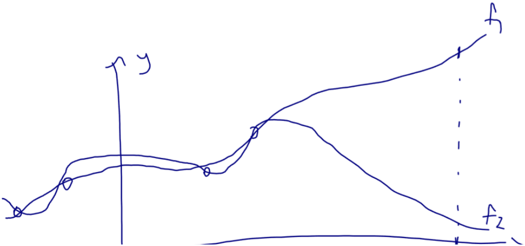
- marginal likelihood of sigma=1 = $p(\mathcal{D}|\sigma = 1, \mathcal{H})$ = 'prob that data came from single Gaussian with param $\sigma = 1$'
- similarly, marginal likelihood of hypothesis = $p(\mathcal{D}|\mathcal{H})$ = 'prob that data came from single Gaussian (with some μ, σ)'

Bayesian Probabilistic Modelling of Functions

Structure

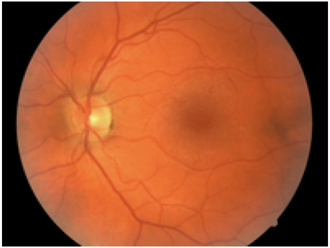
- Why uncertainty over functions
- Linear regression
- Linear basis function regression
- Parametrised basis functions
- Hierarchy of parametrised basis functions (aka neural networks)
- NNs through a probabilistic lens (generative story, prob model, inference, predictions)

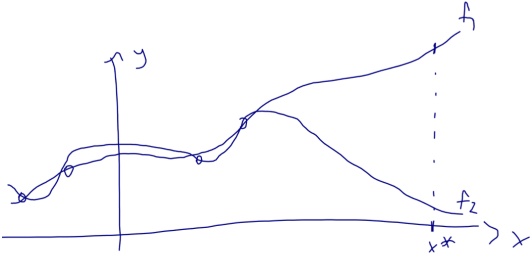




Why uncertainty over functions

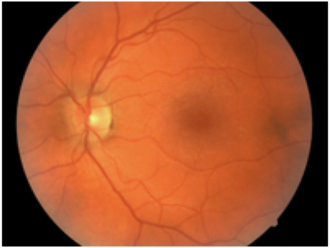
- Going beyond beliefs over discrete set of bets ('heads happened') / scalars (μ)

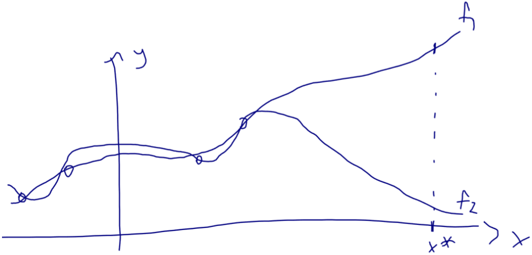




- Want to know **uncertainty** (belief/distribution) of system in its prediction
- Distribution over outputs for **each** input x = dist over *functions*
- First, some preliminaries, some historical context, and notation.

- Going beyond beliefs over discrete set of bets ('heads happened') / scalars (μ)

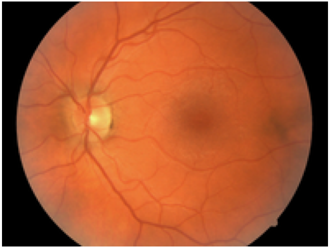


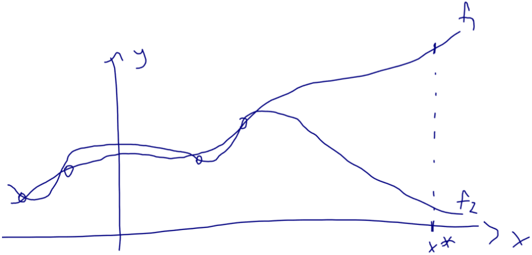


- Want to know **uncertainty** (belief/distribution) of system in its prediction
- Distribution over outputs for **each** input x = dist over *functions*
- First, some preliminaries, some historical context, and notation.

Why uncertainty over functions

- Going beyond beliefs over discrete set of bets ('heads happened') / scalars (μ)





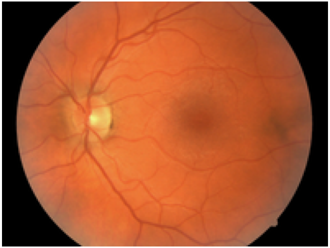
- Want to know **uncertainty** (belief/distribution) of system in its prediction

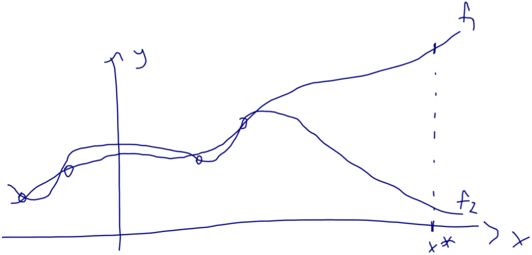
- Distribution over outputs for **each** input x = dist over *functions*

- First, some preliminaries, some historical context, and notation.

Why uncertainty over functions

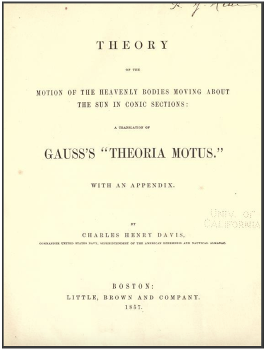
- Going beyond beliefs over discrete set of bets ('heads happened') / scalars (μ)





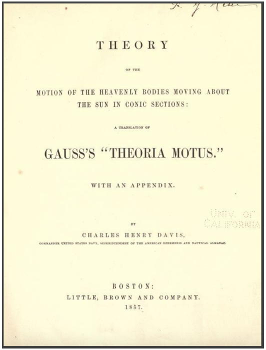
- Want to know **uncertainty** (belief/distribution) of system in its prediction
- Distribution over outputs for **each** input x = dist over *functions*
- First, some preliminaries, some historical context, and notation.

Linear regression [Gauss, 1809]



- Given a set of N input-output pairs $\{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_N, \mathbf{y}_N)\}$
  - eg average number of accidents for different driving speeds
- assumes exists linear func mapping **vectors** $\mathbf{x}_i \in \mathbb{R}^Q$ to $\mathbf{y}_i \in \mathbb{R}^D$ (with $\mathbf{y}_i$ potentially corrupted with observation noise)
- model is linear trans. of inputs: $f(x) = W\mathbf{x} + b$, w W some D by Q matrix over reals, b real vector with D elements
- Different params W, b define different linear trans
  - aim: find params that (eg) minimise $1/N \sum_i ||\mathbf{y}_i - (W\mathbf{x}_i + b)||^2$
- but relation between $\mathbf{x}$ and $\mathbf{y}$ need not be linear...

Linear regression [Gauss, 1809]


THEORY

OF THE

MOTION OF THE HEAVENLY BODIES MOVING ABOUT
THE SUN IN CONIC SECTIONS:

A TRANSLATION OF

GAUSS'S "THEORIA MOTUS."

WITH AN APPENDIX.

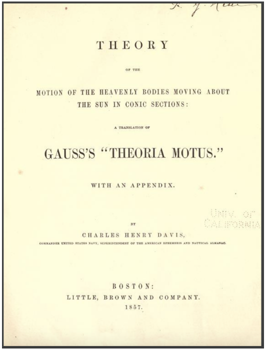
BY

CHARLES HENRY DAVIS,
COMMANDER UNITED STATES NAVY, SUPERINTENDENT OF THE AMERICAN EPHEMERIS AND NAUTICAL ALMANAC.

BOSTON:
LITTLE, BROWN AND COMPANY.
1857.

- Given a set of N input-output pairs $\{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_N, \mathbf{y}_N)\}$
  - eg average number of accidents for different driving speeds
- assumes exists linear func mapping **vectors** $\mathbf{x}_i \in \mathbb{R}^Q$ to $\mathbf{y}_i \in \mathbb{R}^D$ (with $\mathbf{y}_i$ potentially corrupted with observation noise)
- model is linear trans. of inputs: $f(x) = W\mathbf{x} + b$, w W some D by Q matrix over reals, b real vector with D elements
- Different params W, b define different linear trans
  - aim: find params that (eg) minimise $1/N \sum_i ||\mathbf{y}_i - (W\mathbf{x}_i + b)||^2$
- but relation between $\mathbf{x}$ and $\mathbf{y}$ need not be linear...

Linear regression [Gauss, 1809]


THEORY

OF THE

MOTION OF THE HEAVENLY BODIES MOVING ABOUT
THE SUN IN CONIC SECTIONS:

A TRANSLATION OF

GAUSS'S "THEORIA MOTUS."

WITH AN APPENDIX.

BY

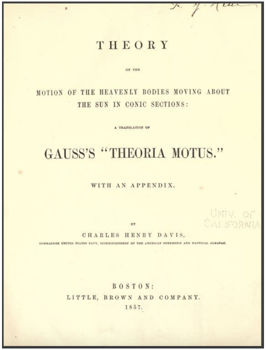
CHARLES HENRY DAVIS,

COMMANDER UNITED STATES NAVY, SUPERINTENDENT OF THE AMERICAN EPHEMERIS AND NAUTICAL ALMANAC.

BOSTON:
LITTLE, BROWN AND COMPANY.
1857.

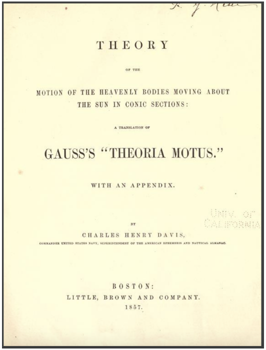
- Given a set of N input-output pairs $\{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_N, \mathbf{y}_N)\}$
  - eg average number of accidents for different driving speeds
- assumes exists linear func mapping **vectors** $\mathbf{x}_i \in \mathbb{R}^Q$ to $\mathbf{y}_i \in \mathbb{R}^D$ (with $\mathbf{y}_i$ potentially corrupted with observation noise)
- model is linear trans. of inputs: $f(x) = W\mathbf{x} + b$, w W some D by Q matrix over reals, b real vector with D elements
- Different params W, b define different linear trans
  - aim: find params that (eg) minimise $1/N \sum_i ||\mathbf{y}_i - (W\mathbf{x}_i + b)||^2$
- but relation between $\mathbf{x}$ and $\mathbf{y}$ need not be linear...

Linear regression [Gauss, 1809]



- Given a set of N input-output pairs $\{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_N, \mathbf{y}_N)\}$
  - eg average number of accidents for different driving speeds
- assumes exists linear func mapping **vectors** $\mathbf{x}_i \in \mathbb{R}^Q$ to $\mathbf{y}_i \in \mathbb{R}^D$ (with $\mathbf{y}_i$ potentially corrupted with observation noise)
- model is linear trans. of inputs: $f(x) = W\mathbf{x} + b$, w W some D by Q matrix over reals, b real vector with D elements
- Different params W, b define different linear trans
  - aim: find params that (eg) minimise $1/N \sum_i ||\mathbf{y}_i - (W\mathbf{x}_i + b)||^2$
- but relation between $\mathbf{x}$ and $\mathbf{y}$ need not be linear...

Linear regression [Gauss, 1809]


THEORY

OF THE

MOTION OF THE HEAVENLY BODIES MOVING ABOUT THE SUN IN CONIC SECTIONS:

A TRANSLATION OF

GAUSS'S "THEORIA MOTUS."

WITH AN APPENDIX.

BY

CHARLES HENRY DAVIS,

BOSTON:
LITTLE, BROWN AND COMPANY.
1857.

- Given a set of N input-output pairs $\{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_N, \mathbf{y}_N)\}$
  - eg average number of accidents for different driving speeds
- assumes exists linear func mapping **vectors** $\mathbf{x}_i \in \mathbb{R}^Q$ to $\mathbf{y}_i \in \mathbb{R}^D$ (with $\mathbf{y}_i$ potentially corrupted with observation noise)
- model is linear trans. of inputs: $f(x) = W\mathbf{x} + b$, w W some D by Q matrix over reals, b real vector with D elements
- Different params W, b define different linear trans
  - aim: find params that (eg) minimise $1/N \sum_i ||\mathbf{y}_i - (W\mathbf{x}_i + b)||^2$
- but relation between $\mathbf{x}$ and $\mathbf{y}$ need not be linear...

UNIVERSITY OF OXFORD

THE APPLICATION OF THE METHOD OF LEAST SQUARES
TO THE INTERPOLATION OF SEQUENCES

By J.D. Gergonne

Translated by Ralph St. John, Bowling Green State University
and S.M. Stigler, University of Wisconsin

- input $\mathbf{x}$ fed through K fixed scalar-valued non-linear trans. $\phi_k(\mathbf{x})$
- trans. ϕ_k are the basis functions; with scalar input x, trans. can be
  - polynomials of degrees k: x^k
  - sinusoidals with various frequencies: $\sin(kx)$
  - wavelets parametrised by k: $\cos(k\pi x)e^{-x^2/2}$
- collect into a **feature vector** $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), ..., \phi_K(\mathbf{x})]$, e.g.

$$\phi(\mathbf{x}) = [\sin(x), \sin(2x)..., \sin(Kx)] \in \mathbb{R}^K$$

- **do linear regression with $\phi(\mathbf{x})$ vector instead of $\mathbf{x}$ itself**

Linear basis function regression [Gergonne, 1815]

THE APPLICATION OF THE METHOD OF LEAST SQUARES
TO THE INTERPOLATION OF SEQUENCES

By J.D. Gergonne

Translated by Ralph St. John, Bowling Green State University
and S.M. Stigler, University of Wisconsin

- input $\mathbf{x}$ fed through K fixed scalar-valued non-linear trans. $\phi_k(\mathbf{x})$
- trans. ϕ_k are the basis functions; with scalar input x, trans. can be
  - polynomials of degrees k: x^k
  - sinusoidals with various frequencies: $\sin(kx)$
  - wavelets parametrised by k: $\cos(k\pi x)e^{-x^2/2}$
- collect into a **feature vector** $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), ..., \phi_K(\mathbf{x})]$, e.g.

$$\phi(\mathbf{x}) = [\sin(x), \sin(2x)..., \sin(Kx)] \in \mathbb{R}^K$$

- do linear regression with $\phi(\mathbf{x})$ vector instead of $\mathbf{x}$ itself

Linear basis function regression [Gergonne, 1815]

UNIVERSITY OF OXFORD

THE APPLICATION OF THE METHOD OF LEAST SQUARES
TO THE INTERPOLATION OF SEQUENCES

By J.D. Gergonne

Translated by Ralph St. John, Bowling Green State University
and S.M. Stigler, University of Wisconsin

- input $\mathbf{x}$ fed through K fixed scalar-valued non-linear trans. $\phi_k(\mathbf{x})$
- trans. ϕ_k are the basis functions; with scalar input x, trans. can be
  - polynomials of degrees k: x^k
  - sinusoidals with various frequencies: $\sin(kx)$
  - wavelets parametrised by k: $\cos(k\pi x)e^{-x^2/2}$
- collect into a **feature vector** $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), ..., \phi_K(\mathbf{x})]$, e.g.

$$\phi(\mathbf{x}) = [\sin(x), \sin(2x)..., \sin(Kx)] \in \mathbb{R}^K$$

- do linear regression with $\phi(\mathbf{x})$ vector instead of $\mathbf{x}$ itself

Linear basis function regression [Gergonne, 1815]

THE APPLICATION OF THE METHOD OF LEAST SQUARES
TO THE INTERPOLATION OF SEQUENCES

By J.D. Gergonne

Translated by Ralph St. John, Bowling Green State University
and S.M. Stigler, University of Wisconsin

- input $\mathbf{x}$ fed through K fixed scalar-valued non-linear trans. $\phi_k(\mathbf{x})$
- trans. ϕ_k are the basis functions; with scalar input x, trans. can be
  - polynomials of degrees k: x^k
  - sinusoidals with various frequencies: $\sin(kx)$
  - wavelets parametrised by k: $\cos(k\pi x)e^{-x^2/2}$
- collect into a **feature vector** $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), ..., \phi_K(\mathbf{x})]$, e.g.

$$\phi(\mathbf{x}) = [\sin(x), \sin(2x)..., \sin(Kx)] \in \mathbb{R}^K$$

- **do linear regression with $\phi(\mathbf{x})$ vector instead of $\mathbf{x}$ itself**

Linear basis function regression [Gergonne, 1815]

UNIVERSITY OF OXFORD

THE APPLICATION OF THE METHOD OF LEAST SQUARES
TO THE INTERPOLATION OF SEQUENCES

By J.D. Gergonne

Translated by Ralph St. John, Bowling Green State University
and S.M. Stigler, University of Wisconsin

- When $\phi_k(\mathbf{x}) := \mathbf{x}_k$ and $K = Q$, basis function regression = linear regression
- basis functions often assumed fixed and orthogonal to each other (optimal combination is sought), e.g. Legendre polynomials
- but need not be fixed and mutually orth. $\to$ param. basis functions

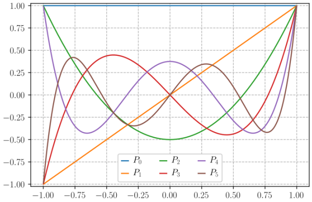
Linear basis function regression [Gergonne, 1815]

THE APPLICATION OF THE METHOD OF LEAST SQUARES
TO THE INTERPOLATION OF SEQUENCES

By J.D. Gergonne

Translated by Ralph St. John, Bowling Green State University
and S.M. Stigler, University of Wisconsin

- When $\phi_k(\mathbf{x}) := \mathbf{x}_k$ and $K = Q$, basis function regression = linear regression
- basis functions often assumed fixed and orthogonal to each other (optimal combination is sought), e.g. Legendre polynomials



Linear basis function regression [Gergonne, 1815]

THE APPLICATION OF THE METHOD OF LEAST SQUARES
TO THE INTERPOLATION OF SEQUENCES

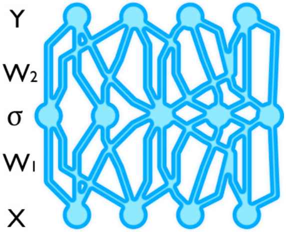
By J.D. Gergonne

Translated by Ralph St. John, Bowling Green State University
and S.M. Stigler, University of Wisconsin

- When $\phi_k(\mathbf{x}) := \mathbf{x}_k$ and $K = Q$, basis function regression = linear regression
- basis functions often assumed fixed and orthogonal to each other (optimal combination is sought), e.g. Legendre polynomials
- but need not be fixed and mutually orth. $\rightarrow$ param. basis functions

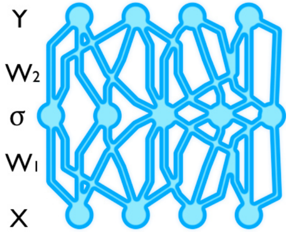
Parametrised basis functions [Bishop, 2006]

- eg basis functions $\phi_k^{w_k,b_k}$ where scalar-valued function ϕ_k is applied to inner-product $w_k^T \mathbf{x} + b_k$
  - ϕ_k often def'd to be identical for all k (only params change)
  - eg $\phi_k(\cdot) = \tanh(\cdot)$, giving $\phi_k^{w_k,b_k}(\mathbf{x}) = \tanh(w_k^T \mathbf{x} + b_k)$
- feature vector = basis functions' outputs = input to linear trans.
- in vector form:
  - W_1 a matrix of dimensions Q by K
  - b_1 a vector with K elements
  - $\phi^{W_1,b_1}(\mathbf{x}) = \phi(W_1 \mathbf{x} + b_1)$
  - W_2 a matrix of dimensions K by D
  - b_2 a vector with D elements
  - model output: $f^{W_1,b_1,W_2,b_2}(\mathbf{x}) = \phi^{W_1,b_1}(\mathbf{x}) W_2 + b_2$
- want to find W_1, b_1, W_2, b_2 that minimise $1/N \sum_i ||\mathbf{y}_i - f^{W_1,b_1,W_2,b_2}(\mathbf{x}_i)||^2$



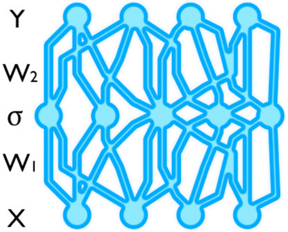
Parametrised basis functions [Bishop, 2006]

- eg basis functions $\phi_k^{w_k,b_k}$ where scalar-valued function ϕ_k is applied to inner-product $w_k^T\mathbf{x} + b_k$
  - ϕ_k often def'd to be identical for all k (only params change)
  - eg $\phi_k(\cdot) = \tanh(\cdot)$, giving $\phi_k^{w_k,b_k}(\mathbf{x}) = \tanh(w_k^T\mathbf{x} + b_k)$
- feature vector = basis functions' outputs = input to linear trans.
- in vector form:
  - W_1 a matrix of dimensions Q by K
  - b_1 a vector with K elements
  - $\phi^{W_1,b_1}(\mathbf{x}) = \phi(W_1\mathbf{x} + b_1)$
  - W_2 a matrix of dimensions K by D
  - b_2 a vector with D elements
  - model output: $f^{W_1,b_1,W_2,b_2}(\mathbf{x}) = \phi^{W_1,b_1}(\mathbf{x})W_2 + b_2$
- want to find W_1, b_1, W_2, b_2 that minimise $1/N\sum_i ||\mathbf{y}_i - f^{W_1,b_1,W_2,b_2}(\mathbf{x}_i)||^2$



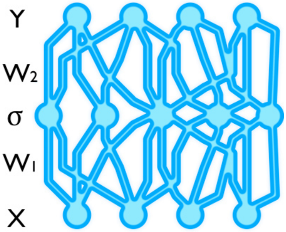
Parametrised basis functions [Bishop, 2006]

- eg basis functions $\phi_k^{w_k,b_k}$ where scalar-valued function ϕ_k is applied to inner-product $w_k^T\mathbf{x} + b_k$
  - ϕ_k often def'd to be identical for all k (only params change)
  - eg $\phi_k(\cdot) = \tanh(\cdot)$, giving $\phi_k^{w_k,b_k}(\mathbf{x}) = \tanh(w_k^T\mathbf{x} + b_k)$
- feature vector = basis functions' outputs = input to linear trans.
- in vector form:
  - W_1 a matrix of dimensions Q by K
  - b_1 a vector with K elements
  - $\phi^{W_1,b_1}(\mathbf{x}) = \phi(W_1\mathbf{x} + b_1)$
  - W_2 a matrix of dimensions K by D
  - b_2 a vector with D elements
  - model output: $f^{W_1,b_1,W_2,b_2}(\mathbf{x}) = \phi^{W_1,b_1}(\mathbf{x})W_2 + b_2$



- want to find W_1, b_1, W_2, b_2 that minimise $1/N\sum_i ||\mathbf{y}_i - f^{W_1,b_1,W_2,b_2}(\mathbf{x}_i)||^2$

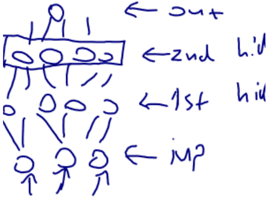
- eg basis functions $\phi_k^{w_k,b_k}$ where scalar-valued function ϕ_k is applied to inner-product $w_k^T\mathbf{x} + b_k$
  - ϕ_k often def'd to be identical for all k (only params change)
  - eg $\phi_k(\cdot) = \tanh(\cdot)$, giving $\phi_k^{w_k,b_k}(\mathbf{x}) = \tanh(w_k^T\mathbf{x} + b_k)$
- feature vector = basis functions' outputs = input to linear trans.
- in vector form:
  - W_1 a matrix of dimensions Q by K
  - b_1 a vector with K elements
  - $\phi^{W_1,b_1}(\mathbf{x}) = \phi(W_1\mathbf{x} + b_1)$
  - W_2 a matrix of dimensions K by D
  - b_2 a vector with D elements
  - model output: $f^{W_1,b_1,W_2,b_2}(\mathbf{x}) = \phi^{W_1,b_1}(\mathbf{x})W_2 + b_2$



- want to find W_1, b_1, W_2, b_2 that minimise $1/N\sum_i ||\mathbf{y}_i - f^{W_1,b_1,W_2,b_2}(\mathbf{x}_i)||^2$

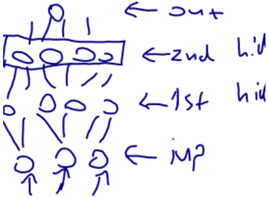
Hierarchy of parametrised basis functions [Rumelhart et al., 1985]

- called "NNs" for historical reasons
- layers
  - ='feature vectors' in hierarchy
  - linear trans. = 'inner product' layer = 'fully connected' layer
  - 'input layer', 'output layer', 'hidden layers'
  - trans. matrix = weight matrix = W; intercept = bias = b
- **units**
  - elements in a layer
- **feature vector** (overloaded term)
  - often refers to the penultimate layer (at top of model just before softmax / last linear trans.)
  - denote feature vector



Hierarchy of parametrised basis functions [Rumelhart et al., 1985]

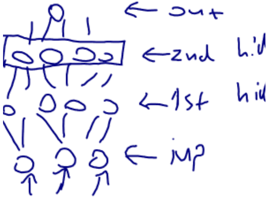
- called “NNs” for historical reasons
- **layers**
  - ='feature vectors' in hierarchy
  - linear trans. = ‘inner product’ layer = ‘fully connected’ layer
  - ‘input layer’, ‘output layer’, ‘hidden layers’
  - trans. matrix = weight matrix = W; intercept = bias = b
- **units**
  - elements in a layer
- **feature vector** (overloaded term)
  - often refers to the penultimate layer (at top of model just before softmax / last linear trans.)
  - denote feature vector



Hierarchy of parametrised basis functions

Hierarchy of parametrised basis functions [Rumelhart et al., 1985]

- called “NNs” for historical reasons
- **layers**
  - =‘feature vectors’ in hierarchy
  - linear trans. = ‘inner product’ layer = ‘fully connected’ layer
  - ‘input layer’, ‘output layer’, ‘hidden layers’
  - trans. matrix = weight matrix = W; intercept = bias = b
- **units**
  - elements in a layer

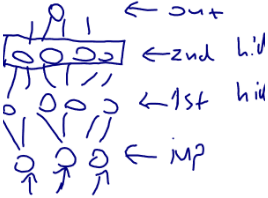


- **feature vector** (overloaded term)
  - often refers to the penultimate layer (at top of model just before softmax / last linear trans.)
  - denote feature vector

Hierarchy of parametrised basis functions

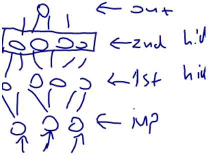
Hierarchy of parametrised basis functions [Rumelhart et al., 1985]

- called "NNs" for historical reasons
- **layers**
- **units**
- **feature vector** (overloaded term)
  - often refers to the penultimate layer (at top of model just before softmax / last linear trans.)
  - denote feature vector $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), .., \phi_K(\mathbf{x})]$ with K units (a K by 1 vector)
  - denote feature matrix $\Phi(\mathbf{X}) = [\phi(\mathbf{x}_1)^T, ..., \phi(\mathbf{x}_N)^T]$, N by K matrix



Hierarchy of parametrised basis functions

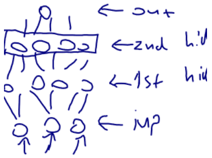
- **regression**
  - compose multiple basis function layers into a regression model
  - result of last trans. also called "model output"; often no non-linearity here



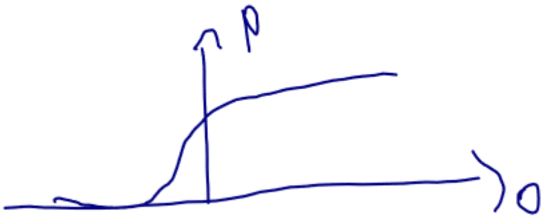
- classification
  - further compose a **softmax** function at the end; also called "logistic" for 2 classes
  - "squashes" its input → probability vector; prob vector also called model output / softmax vector / softmax layer
- "building blocks"
  - layers are simple
  - modularity in layer composition → versatility of deep models
  - many engineers work in field → lots of tools that scale well

Hierarchy of parametrised basis functions

- **regression**
  - compose multiple basis function layers into a regression model
  - result of last trans. also called "model output"; often no non-linearity here

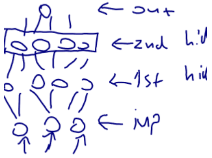


- **classification**
  - further compose a **softmax** function at the end; also called "logistic" for 2 classes
  - "squashes" its input → probability vector; prob vector also called model output / softmax vector / softmax layer



Hierarchy of parametrised basis functions

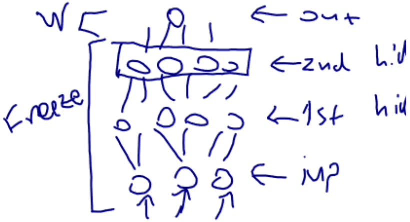
- **regression**
  - compose multiple basis function layers into a regression model
  - result of last trans. also called "model output"; often no non-linearity here



- **classification**
  - further compose a **softmax** function at the end; also called "logistic" for 2 classes
  - "squashes" its input → probability vector; prob vector also called model output / softmax vector / softmax layer
- "building blocks"
  - layers are simple
  - modularity in layer composition → versatility of deep models
  - many engineers work in field → lots of tools that scale well

Assumptions **for the moment**

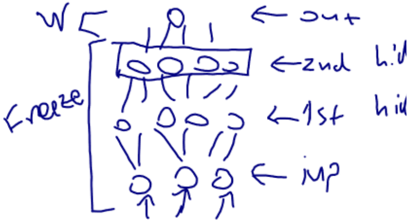
- we'll use deep nets, and denote W to be the weight matrix of the last layer and b the bias of last layer
- (for the moment) look only at last layer W, everything else fixed – ie weights other than W do not change
  - later we'll worry about other layers



- assume that y is scalar
  - so W is K by 1 (all vectors are forever column vectors)
  - write w_k for the k'th elem

Assumptions **for the moment**

- we'll use deep nets, and denote W to be the weight matrix of the last layer and b the bias of last layer
- (for the moment) look only at last layer W, everything else fixed – ie weights other than W do not change
  - later we'll worry about other layers



- assume that y is scalar
  - so W is K by 1 (all vectors are forever column vectors)
  - write w_k for the k'th elem

Assumptions **for the moment**

- we'll use deep nets, and denote W to be the weight matrix of the last layer and b the bias of last layer
- (for the moment) look only at last layer W, everything else fixed – ie weights other than W do not change
  - later we'll worry about other layers
- assume that y is scalar
  - so W is K by 1 (all vectors are forever column vectors)
  - write w_k for the k'th elem
- assume that output layer's b is zero (or, obs y's are normalised)
  - both will simplify derivations here (but pose no difficulty otherwise)
- then $f^W(\mathbf{x}) = \sum w_k \phi_k(\mathbf{x}) = W^T \phi(x)$ with $\phi(x)$ a 'frozen' feature vec for some NN
- some notation you'll need to remember...
  $\mathbf{X}, x, N, \mathbf{x}_n, Q, D, K, \mathcal{D} = \{(\mathbf{x}_1, y_1), .., (\mathbf{x}_N, y_N)\} = \mathbf{X}, \mathbf{Y}$

Assumptions for the moment

- we'll use deep nets, and denote W to be the weight matrix of the last layer and b the bias of last layer
- (for the moment) look only at last layer W, everything else fixed – ie weights other than W do not change
  - later we'll worry about other layers
- assume that y is scalar
  - so W is K by 1 (all vectors are forever column vectors)
  - write w_k for the k'th elem
- assume that output layer's b is zero (or, obs y's are normalised)
  - both will simplify derivations here (but pose no difficulty otherwise)
- then $f^W(\mathbf{x}) = \sum w_k \phi_k(\mathbf{x}) = W^T \phi(x)$ with $\phi(x)$ a 'frozen' feature vec for some NN
- some notation you'll need to remember...
  $\mathbf{X}, x, N, \mathbf{x}_n, Q, D, K, \mathcal{D} = \{(\mathbf{x}_1, y_1), .., (\mathbf{x}_N, y_N)\} = \mathbf{X}, \mathbf{Y}$

Assumptions for the moment

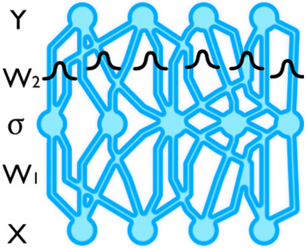
- we'll use deep nets, and denote W to be the weight matrix of the last layer and b the bias of last layer
- (for the moment) look only at last layer W, everything else fixed – ie weights other than W do not change
  - later we'll worry about other layers
- assume that y is scalar
  - so W is K by 1 (all vectors are forever column vectors)
  - write w_k for the k'th elem
- assume that output layer's b is zero (or, obs y's are normalised)
  - both will simplify derivations here (but pose no difficulty otherwise)
- then $f^W(\mathbf{x}) = \sum w_k \phi_k(\mathbf{x}) = W^T \phi(x)$ with $\phi(x)$ a 'frozen' feature vec for some NN
- some notation you'll need to remember... $\mathbf{X}, x, N, \mathbf{x}_n, Q, D, K, \mathcal{D} = \{(\mathbf{x}_1, y_1), .., (\mathbf{x}_N, y_N)\} = \mathbf{X}, \mathbf{Y}$

Assumptions **for the moment**

- we'll use deep nets, and denote W to be the weight matrix of the last layer and b the bias of last layer
- (for the moment) look only at last layer W, everything else fixed – ie weights other than W do not change
  - later we'll worry about other layers
- assume that y is scalar
  - so W is K by 1 (all vectors are forever column vectors)
  - write w_k for the k'th elem
- assume that output layer's b is zero (or, obs y's are normalised)
  - both will simplify derivations here (but pose no difficulty otherwise)
- then $f^W(\mathbf{x}) = \sum w_k \phi_k(\mathbf{x}) = W^T \phi(x)$ with $\phi(x)$ a 'frozen' feature vec for some NN
- some notation you'll need to remember...
  $\mathbf{X}, x, N, \mathbf{x}_n, Q, D, K, \mathcal{D} = \{(\mathbf{x}_1, y_1), .., (\mathbf{x}_N, y_N)\} = \mathbf{X}, \mathbf{Y}$

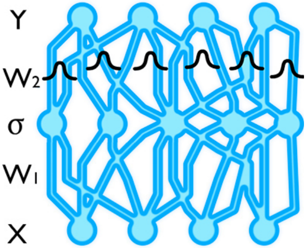
Want to put dist over functions..

- difficult to put belief over funcs., but easy to put over NN params
- assumptions for the moment: our data was generated from the fixed ϕ (NN) using *some* W (which we want to infer)



Want to put dist over functions..

- difficult to put belief over funcs., but easy to put over NN params
- assumptions for the moment: our data was generated from the fixed ϕ (NN) using *some* W (which we want to infer)



Generative story [what we assume about the data]

- Nature chose W which def's a func: $f^W(x) := W^T\phi(x)$
- generated **func. values** with inputs $x_1, .., x_N$: $f_n := f^W(x_n)$
- corrupted func. values with noise [also called "obs noise"]
  $y_n := f_n + \epsilon_n, \ \epsilon_n \sim \mathcal{N}(0, \sigma^2)$ [additive Gaussian noise w param σ]
- we're given **observations** $\{(x_1, y_1), ..., (x_N, y_N)\}$ and $\sigma = 1$

Model

- qs
  - how can we find function value f^* for a new x^*?
  - how can we find our confidence in this prediction?
  - $\rightarrow$ 'everything follows from the laws of probability theory'
- we build a **model**:
  - put prior dist over params W

$$p(W) = \mathcal{N}(W; 0_K, s^2 I_K)$$

  - likelihood [conditioned on W generate obs by adding gaussian noise]

$$p(y|W, x) = \mathcal{N}(y; W^T \phi(x), \sigma^2)$$

- prior belief "$w_k \sim \mathcal{N}(0, 1)$" means that **funcs** are likely to be smooth, vs "$w_k \in \mathcal{N}(0, 100)$" means funcs likely to be erratic; eg, visualising func draws from prior:
- we want to infer W (find dist over W given $\mathcal{D}$)

- qs
  - how can we find function value f^* for a new x^*?
  - how can we find our confidence in this prediction?
  - $\rightarrow$ 'everything follows from the laws of probability theory'
- we build a **model**:
  - put prior dist over params W

$$p(W) = \mathcal{N}(W; 0_K, s^2 I_K)$$

  - likelihood [conditioned on W generate obs by adding gaussian noise]

$$p(y|W, x) = \mathcal{N}(y; W^T \phi(x), \sigma^2)$$

- prior belief "$w_k \sim \mathcal{N}(0, 1)$" means that **funcs** are likely to be smooth, vs "$w_k \in \mathcal{N}(0, 100)$" means funcs likely to be erratic; eg, visualising func draws from prior:
- we want to infer W (find dist over W given $\mathcal{D}$)

Model

- qs
  - how can we find function value f^* for a new x^*?
  - how can we find our confidence in this prediction?
  - $\rightarrow$ 'everything follows from the laws of probability theory'

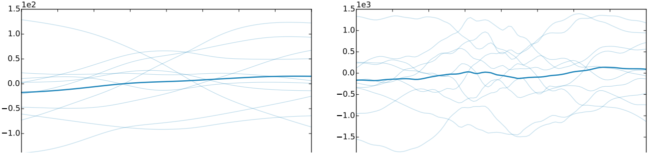
- we build a **model**:
  - put prior dist over params W

$$p(W) = \mathcal{N}(W; 0_K, s^2 I_K)$$

  - likelihood [conditioned on W generate obs by adding gaussian noise]

$$p(y|W, x) = \mathcal{N}(y; W^T \phi(x), \sigma^2)$$

- prior belief "$w_k \sim \mathcal{N}(0, 1)$" means that **funcs** are likely to be smooth, vs "$w_k \in \mathcal{N}(0, 100)$" means funcs likely to be erratic; eg, visualising func draws from prior:



- qs
  - how can we find function value f^* for a new x^*?
  - how can we find our confidence in this prediction?
  - $\rightarrow$ 'everything follows from the laws of probability theory'
- we build a **model**:
  - put prior dist over params W

$$p(W) = \mathcal{N}(W; 0_K, s^2 I_K)$$

  - likelihood [conditioned on W generate obs by adding gaussian noise]

$$p(y|W, x) = \mathcal{N}(y; W^T \phi(x), \sigma^2)$$

- prior belief "$w_k \sim \mathcal{N}(0, 1)$" means that **funcs** are likely to be smooth, vs "$w_k \in \mathcal{N}(0, 100)$" means funcs likely to be erratic; eg, visualising func draws from prior:
- we want to infer W (find dist over W given $\mathcal{D}$)

Analytic inference w functions [new technique!]

We know that the prior and likelihood are Gaussians, therefore the posterior prob over W must be Gaussian too (conjugacy). We'll complete the squares in the exponents to find it:

$p(w \mid x, y)$

$= p(y \mid x, w)\, \overset{p(w)}{p(w \mid \underline{\underline{x}})} \,/\, \underline{\underline{p(y \mid x)}}$

[whiteboard]

We know that the prior and likelihood are Gaussians, therefore the posterior prob over W must be Gaussian too (conjugacy). We'll complete the squares in the exponents to find it:

$p(w \mid x, y)$

$= p(y \mid x, w)\, \overset{p(w)}{p(w \mid \underline{x})} \,/\, \underline{p(y \mid x)}$

[whiteboard]

- posterior variance

$$\Sigma' = (\sigma^{-2} \sum_n (\phi(x_n)\phi(x_n)^T) + s^{-2} I_K)^{-1}$$

  and in vector form: $(\sigma^{-2}\Phi(\mathbf{X})^T\Phi(\mathbf{X}) + s^{-2} I_K)^{-1}$

- posterior mean

$$\mu' = \Sigma' \sigma^{-2} \sum_n (y_n \phi(x_n))$$

  and in vector form: $\Sigma' \sigma^{-2} \Phi(\mathbf{X})^T \mathbf{Y}$

Analytic predictions with functions

How do we predict function values y^* for new x^*?

- There's only one correct way – use prob theory to perform preds!

$$p(y^*|x^*, X, Y) \qquad \text{this is the } \textbf{predictive} \text{ dist (we'll use it a lot)}$$
$$= \int p(y^*, W|x^*, X, Y)\mathrm{d}W \qquad \text{sum rule}$$
$$= \int p(y^*|x^*, W, X, Y)p(W|X, Y)\mathrm{d}W \qquad \text{product rule}$$
$$= \int p(y^*|x^*, W)p(W|X, Y)\mathrm{d}W \qquad \text{model assumptions}$$

- how to eval? [a new technique!]
  - likelihood $p(y^*|x^*, W)$ is Gaussian
  - posterior $p(W|X, Y)$ is Gaussian (from above)
  - so *predictive* $p(y^*|x^*, X, Y)$ is Gaussian..

Analytic predictions with functions

We know that the likelihood and posterior are Gaussians, therefore the predictive prob over y^* must be Gaussian too (conjugacy). We'll use moment matching to find it:

$$p(y^* | x^*, D) = \mathcal{N}(\mu^*, \Sigma^*)$$

$$\mu^* = E_{p(y^*|x^*, D)}[y^*]$$

[whiteboard]

Analytic predictions with functions

We know that the likelihood and posterior are Gaussians, therefore the predictive prob over y^* must be Gaussian too (conjugacy). We'll use moment matching to find it:

$p(y^* | x^*, D) = \mathcal{N}(\mu^*, \Sigma^*)$

$\mu^* = E_{p(y^*|x^*, D)}[y^*]$

- **Exercise** (10min): Derive the variance of the predictive distribution (predictive variance) (hint: use the identity $Var(z) = E[z^T z] - E[z]^T E[z]$ with simple manipulations)

Useful resources: yr.gl/udl101

Analytic predictions with functions (Exercise)

- Predictive variance (hint: use the identity $Var(z) = E[z^T z] - E[z]^T E[z]$ with simple manipulations)

$$Var(y^*) = E_{p(y^*|..)}[y^{*T} y^*] - \mu^{*T} \mu^*$$

$$= \sigma^2 + \phi(x^*)^T \Sigma' \phi(x^*)$$

Questions & discussion

SLIDES

Slide decks from the talks.

SLIDE DECK 1

SLIDE DECK 2

DEMO

Demoes mentioned in the slides

UNCERTAINTY PLAYGROUND

UNCERTAINTY VISUALISATION

RECAP

A quick recap of useful stuff.

GAUSSIANS RECAP

NOTATION

Notation used in the slides:.

MORE STUFF

OATML

Temporary page!

LaTeX was unable to guess the total number of pages correctly. A
was some unprocessed data that should have been added to the
page this extra page has been added to receive it.
If you rerun the document (without altering it) this surplus page
away, because LaTeX now knows how many pages to expect for t
document.