COMP 551 - Applied Machine Learning Lecture 23 — Bayesian regression

William L. Hamilton

(with slides and content from Joelle Pineau)

* Unless otherwise noted, all material posted for this course are copyright of the instructor, and cannot be reused or reposted without the instructor's written permission.

Some logistics

- Midterm and MiniProject 3 grades will be released by Tuesday, April 9th.
- We will have an extra office hour session at 4pm in MC 321 on April 9th for people to view their exams and the solutions.
- Submit your course reviews: https://horizon.mcgill.ca/pban1/twbkwbis.P_WWWLogin?ret_code=f
-

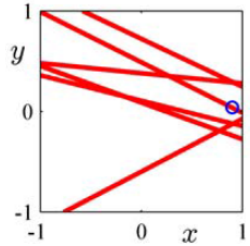
Recall: Bayesian probabilities

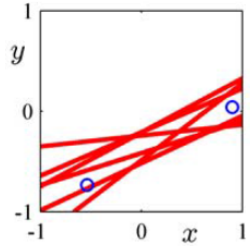
- An example from regression
- Given few noisy data points, multiple parameter values are possible
- Can we quantify uncertainty over our parameters using probabilities?

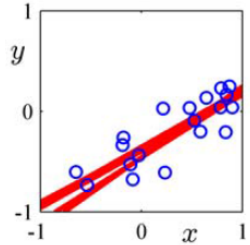
- I.e. given a dataset: $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$

  and some model with weights $\mathbf{w}$, can we find:

$$p(\mathbf{w}|\mathcal{D})$$







Copyright C.M. Bishop, PRML

Recall: Bayesian terminology

$p(\mathcal{D}|\mathbf{w})$

- Likelihood: our model of the data. Given our weights, how do we assign probabilities to dataset examples?

$p(\mathbf{w})$

- Prior: before we see any data, what do we think about our parameters?

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$$

- Posterior: our distribution over weights, given the data we've observed and our prior

$p(\mathcal{D})$

- Marginal likelihood: also called the normalization constant. Does not depend on w, so not usually calculated explicitly

Recall: Conjugate priors

- A *prior* $p(\mathbf{w})$ is *conjugate* to a *likelihood* function $p(\mathcal{D}|\mathbf{w})$ if the posterior is in the same family as the prior

- In other words, if *prior * likelihood* gives you the same form as the prior with different parameters, it's a conjugate prior
  - Ex 1: the Gaussian distribution is a conjugate prior to a Gaussian likelihood
  - Ex 2: the Beta distribution is conjugate to a Bernoulli likelihood

- **Why?** *Want simple form for our posterior!* Don't want it to get more complicated every time you add more data

Bayesian linear regression

- Previous examples (coin flip, learning the mean of a Gaussian) only had outputs $\mathbf{y}$, no inputs $\mathbf{x}$
- How can we learn to make predictions that are input-dependent?
- Can use an extension of linear regression: Bayesian linear regression

Recall: Algorithms for Bayesian inference

- Given a dataset $\mathcal{D}$, how do we make predictions for a new input?

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$$

- Step 1: Define a model that represents your data (the **likelihood**): $p(\mathcal{D}|\mathbf{w})$
- Step 2: Define a **prior** over model parameters: $p(\mathbf{w})$
- Step 3: Calculate **posterior** using Bayes' rule: $p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$
- Step 4: Make **prediction** by integrating over model parameters:

$$p(y^*|\mathbf{x}^*, \mathcal{D}) = \int_{\mathbb{R}^N} p(\mathbf{w}|\mathcal{D})p(y^*|\mathbf{x}^*, \mathbf{w})d\mathbf{w}$$

Bayesian linear regression

- We take a *specific form of the likelihood and the prior*:
  - Step 1: Likelihood

$$p(y|\mathbf{x}, \mathbf{w}) = \mathcal{N}(\mathbf{w}^T\mathbf{x}, \sigma^2)$$

Output *y* close to learned linear function **w*x** , with some noise

  - Step 2: Conjugate prior

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, \alpha^{-1}\mathbf{I})$$

Prefer small weights. (assuming no other info)

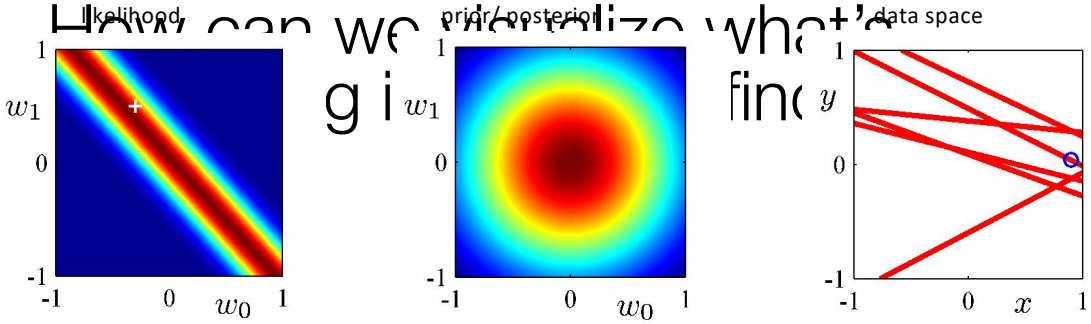
  - Prior precision α and noise variance σ^2 considered known
  - Linear regression where we **learn a distribution over the parameters**

Visualizing inference

- Start with simple example (one feature x):
- How can we visualize what's going on during inference?

$y = w_0 + w_1 x + \epsilon$

$p(\mathbf{w}|\mathcal{D})$



For different w0, w1, how likely is this data point?

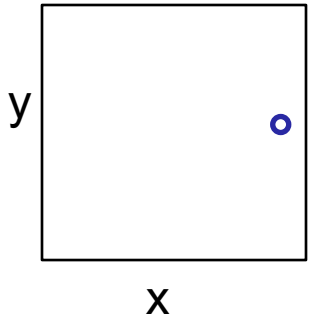
How likely are different (w0, w1) given data so far?

Shows data points and sample functions for data so far

Copyright C.M. Bishop, PRML

Visualizing inference

- Goal: fit lines $y = w_0 + w_1 x + \epsilon$
- Bayes theorem: $p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$

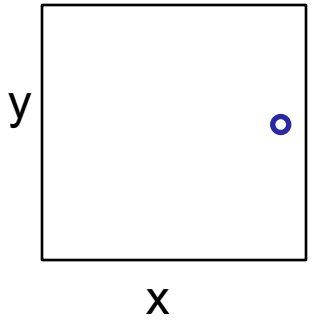


Visualizing inference

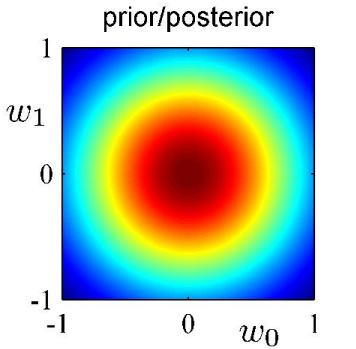
- Goal: fit lines $y = w_0 + w_1 x + \epsilon$

What prior?

- Bayes theorem: $p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$



- Similar to ridge regression, expect good **w** to be small



Copyright C.M. Bishop, PRML

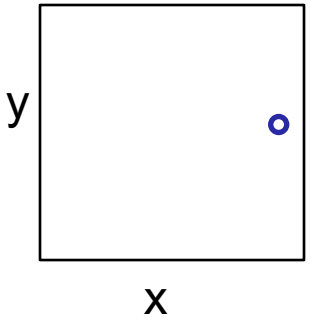
Visualizing inference

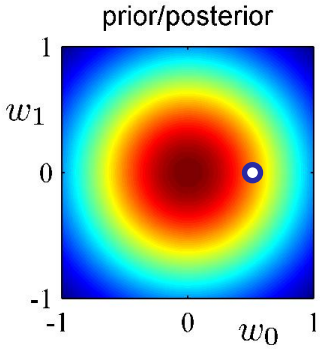
- Goal: fit lines $y = w_0 + w_1 x + \epsilon$

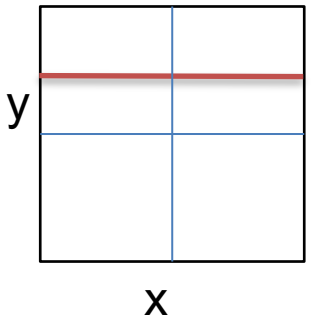
What prior?

- Bayes theorem: $p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$

- Similar to ridge regression, expect good **w** to be small







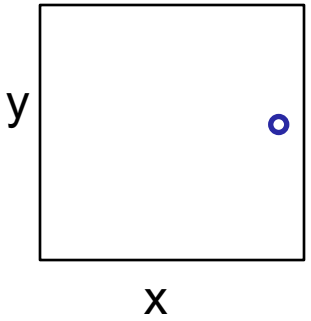
Visualizing inference

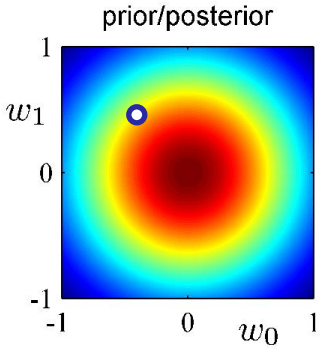
- Goal: fit lines $y = w_0 + w_1 x + \epsilon$

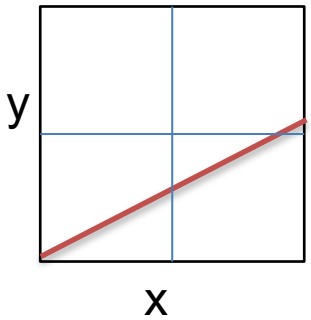
What prior?

- Bayes theorem: $p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$

- Similar to ridge regression, expect good **w** to be small







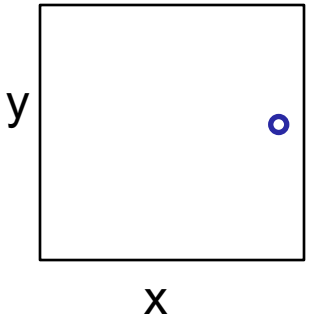
Visualizing inference

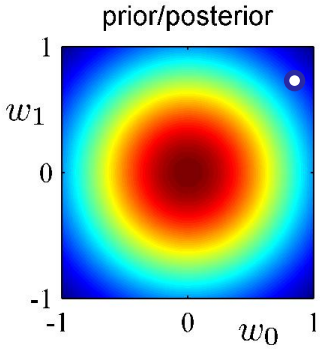
- Goal: fit lines $y = w_0 + w_1 x + \epsilon$

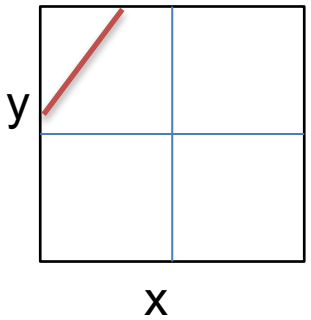
What prior?

- Bayes theorem: $p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$

- Similar to ridge regression, expect good **w** to be small



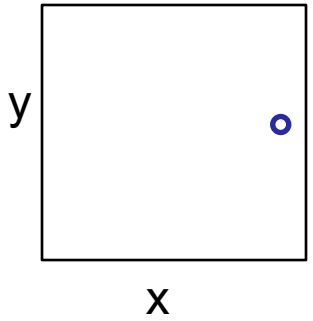




Visualizing inference

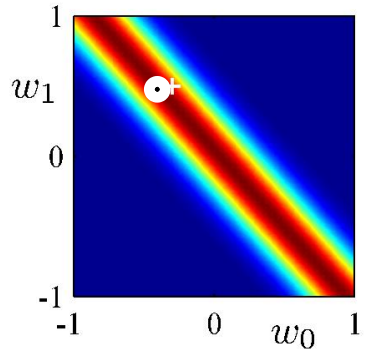
- Goal: fit lines $y = w_0 + w_1 x + \epsilon$

What about the likelihood?

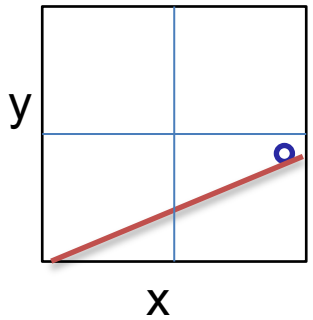


- Bayes theorem: $p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$

- Good lines should pass 'close by' datapoint



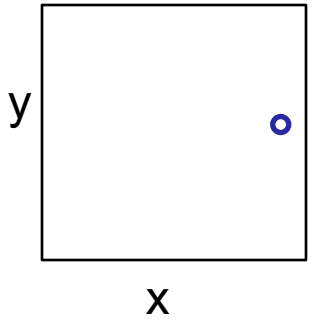
Copyright C.M. Bishop, PRML



Visualizing inference

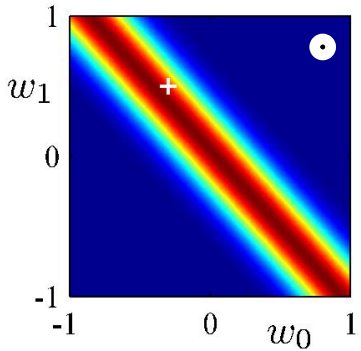
- Goal: fit lines $y = w_0 + w_1 x + \epsilon$

What about the likelihood?

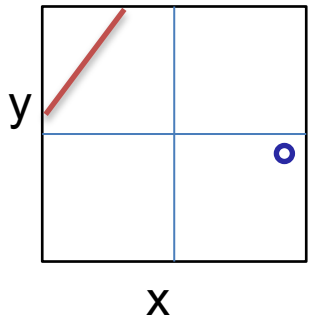


- Bayes theorem: $p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$

- Good lines should pass 'close by' datapoint

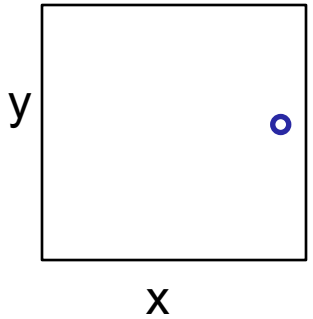


Copyright C.M. Bishop, PRML



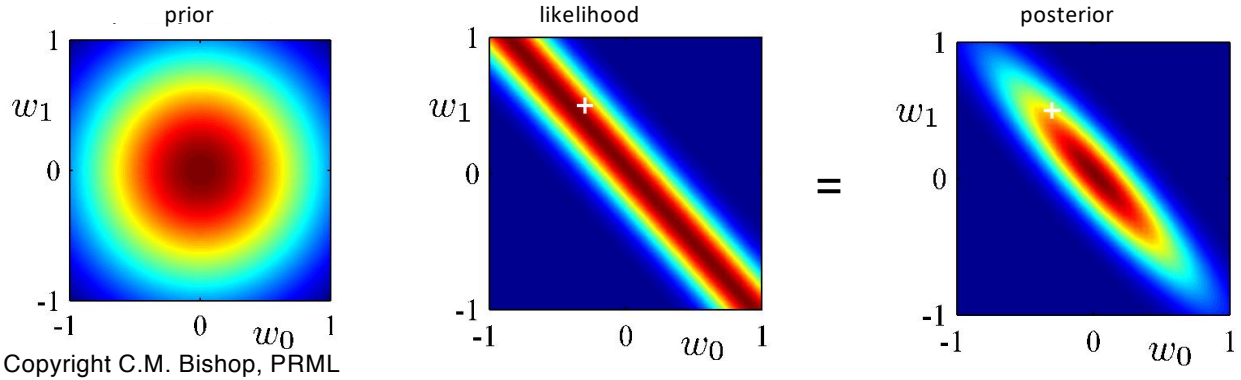
Visualizing inference

- Goal: fit lines $y = w_0 + w_1 x + \epsilon$



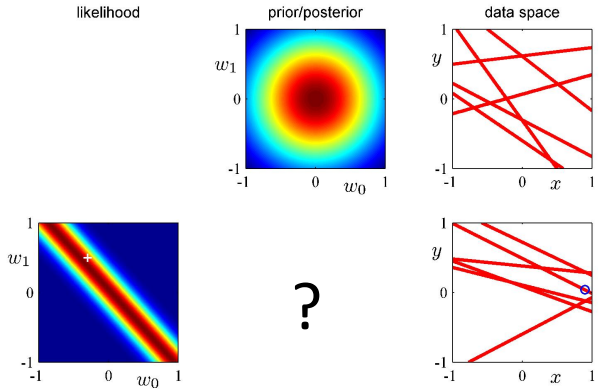
- Bayes theorem: $p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$

- For all values of **w**, multiply prior and likelihood (and re-normalize)



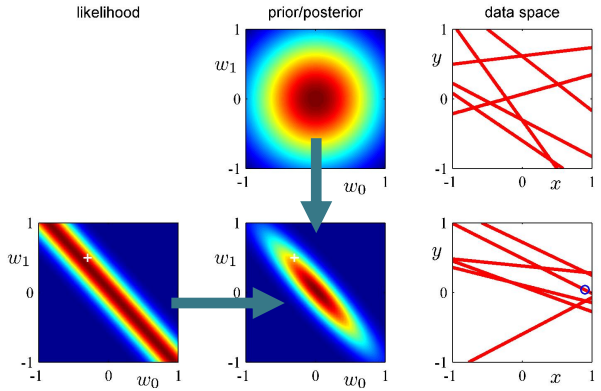
Copyright C.M. Bishop, PRML

Bayesian linear regression: inference



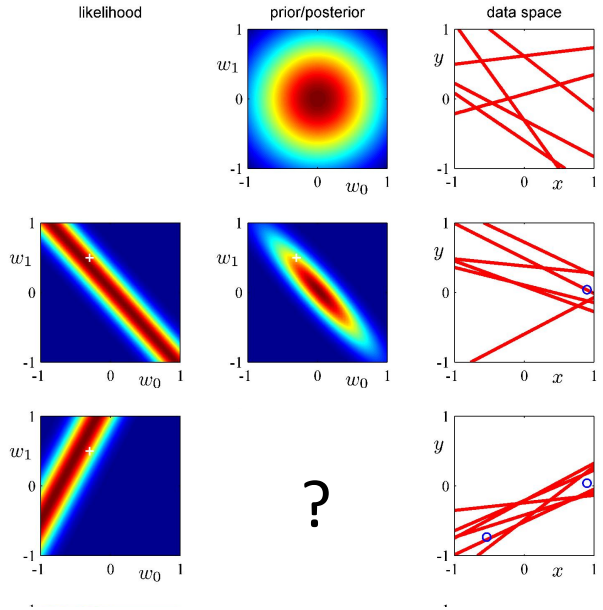
Copyright C.M. Bishop, PRML

Bayesian linear regression: inference



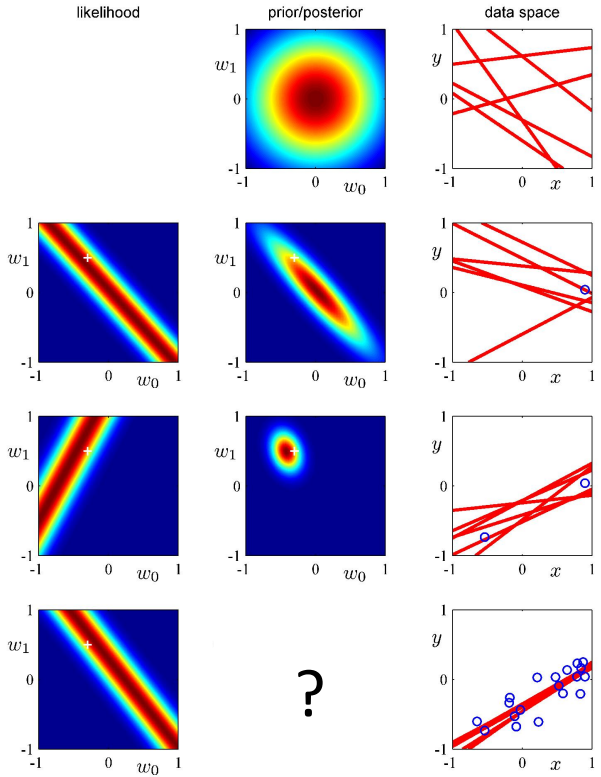
Copyright C.M. Bishop, PRML

Bayesian linear regression: inference



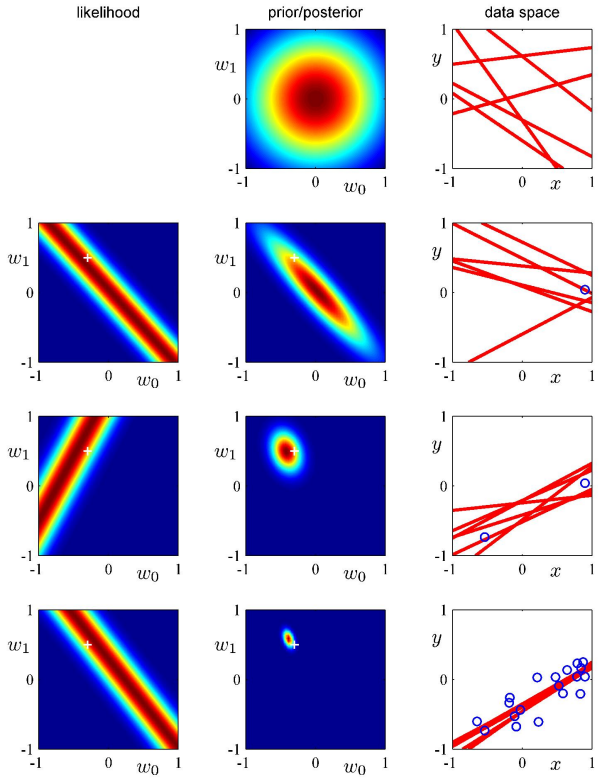
Copyright C.M. Bishop, PRML

Bayesian linear regression: inference



Copyright C.M. Bishop, PRML

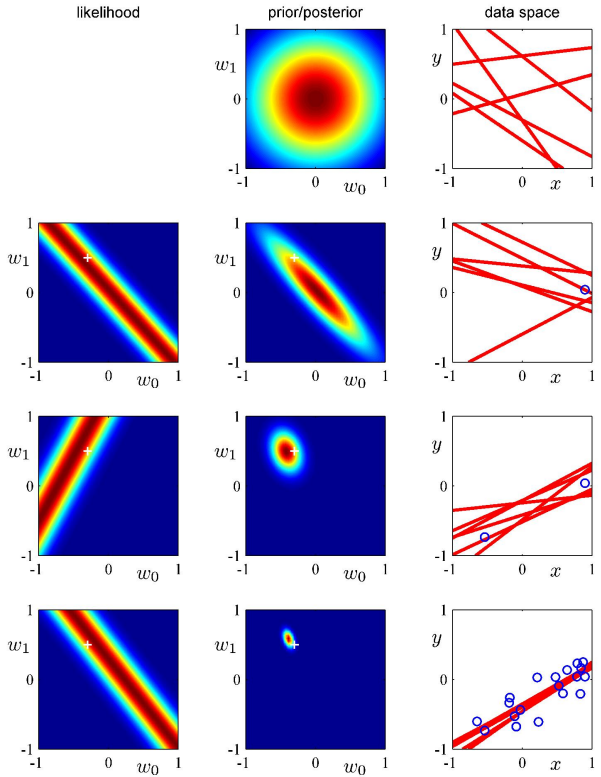
Bayesian linear regression: inference



Copyright C.M. Bishop, PRML

Bayesian linear regression: inference

As new data points are added, posterior converges on true value of parameters



Copyright C.M. Bishop, PRML

Step 3: calculate posterior

- Can calculate posterior by multiplying prior and likelihood:

$$p(\mathbf{w}|\mathcal{D}) = \mathcal{N}(\sigma^{-2}\mathbf{S}_N\mathbf{X}^T\mathbf{y}, \mathbf{S}_N)$$

$$\mathbf{S}_N = (\alpha\mathbf{I} + \sigma^{-2}\mathbf{X}^T\mathbf{X})^{-1}$$

(derivation similar to case with no inputs — slide 306 of lecture 18. Full derivation in section 3.3 of Bishop)

- $\mathbf{X}$ has one input per row, $\mathbf{y}$ has one target output per row

- If prior precision α goes to 0, mean becomes maximum likelihood solution (ordinary linear regression)
- Infinitely wide likelihood variance σ^2 or 0 datapoints, means distribution reduces to prior

Aside: finding the MAP

- We can investigate the maximum of the posterior (MAP)
- Log-transform posterior: log is sum of prior + likelihood

$$
\begin{aligned}
& \max \log p(\mathbf{w}|\mathbf{y}) \\
= & \\
& \max -\frac{\sigma^{-2}}{2} \sum_{n=1}^{N} (y_n - \mathbf{w}^T \mathbf{x}_n)^2 - \frac{\alpha}{2} \mathbf{w}^T \mathbf{w} + \text{const.}
\end{aligned}
$$

Aside: finding the MAP

- We can investigate the maximum value of the posterior (MAP)
- Calculate in log space: *log posterior = log prior + log likelihood*

$$\max \log p(\mathbf{w}|\mathbf{y})$$

$$= \max -\frac{\sigma^{-2}}{2}\sum_{n=1}^{N}(y_n - \mathbf{w}^T\mathbf{x}_n)^2 - \frac{\alpha}{2}\mathbf{w}^T\mathbf{w} + \text{const.}$$

Recall:

$$\min \sum_{n=1}^{N}(y_n - \mathbf{w}^T\mathbf{x}_n)^2 + \lambda\mathbf{w}^T\mathbf{w}$$

Ridge regression, (I.e., L2-regularized linear regression)

- Same objective function as for ridge regression!
- Penalty term: $\lambda = \alpha\sigma^2$

prior precision

likelihood variance

Note: since posterior is Gaussian, MAP = mean of posterior

Step 4: prediction

- Prediction for new datapoint:

$$p(y^*|\mathbf{x}^*, \mathcal{D}) = \int_{\mathbb{R}^N} p(\mathbf{w}|\mathcal{D}) p(y^*|\mathbf{x}^*, \mathbf{w}) d\mathbf{w}$$

- For Gaussians, can compute solution analytically:

$$p(y^*|\mathcal{D}) = \mathcal{N}(\sigma^{-2}\mathbf{x}^{*T}\mathbf{S}_N\mathbf{X}^T\mathbf{y}, \sigma^2 + \mathbf{x}^T\mathbf{S}_N\mathbf{x})$$

mean from before

from weight uncertainty

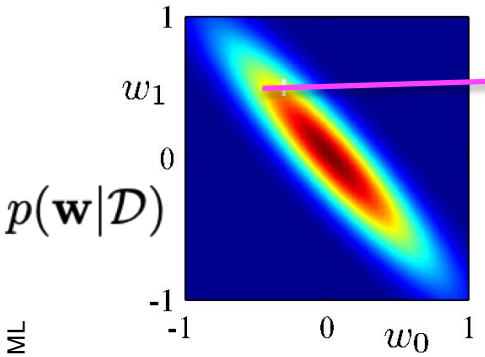
new input

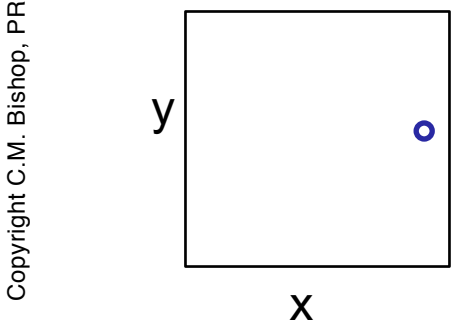
from observation noise

- Variance tends to go down with more data until it reaches σ^2

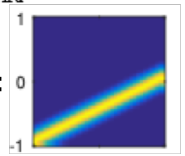
Step 4: prediction

- Every **w** makes a prediction, weighted by posterior





$$\int_{\mathbb{R}^N} p(\mathbf{w}|\mathcal{D})p(y^*|\mathbf{x}^*,\mathbf{w})d\mathbf{w}$$

= 

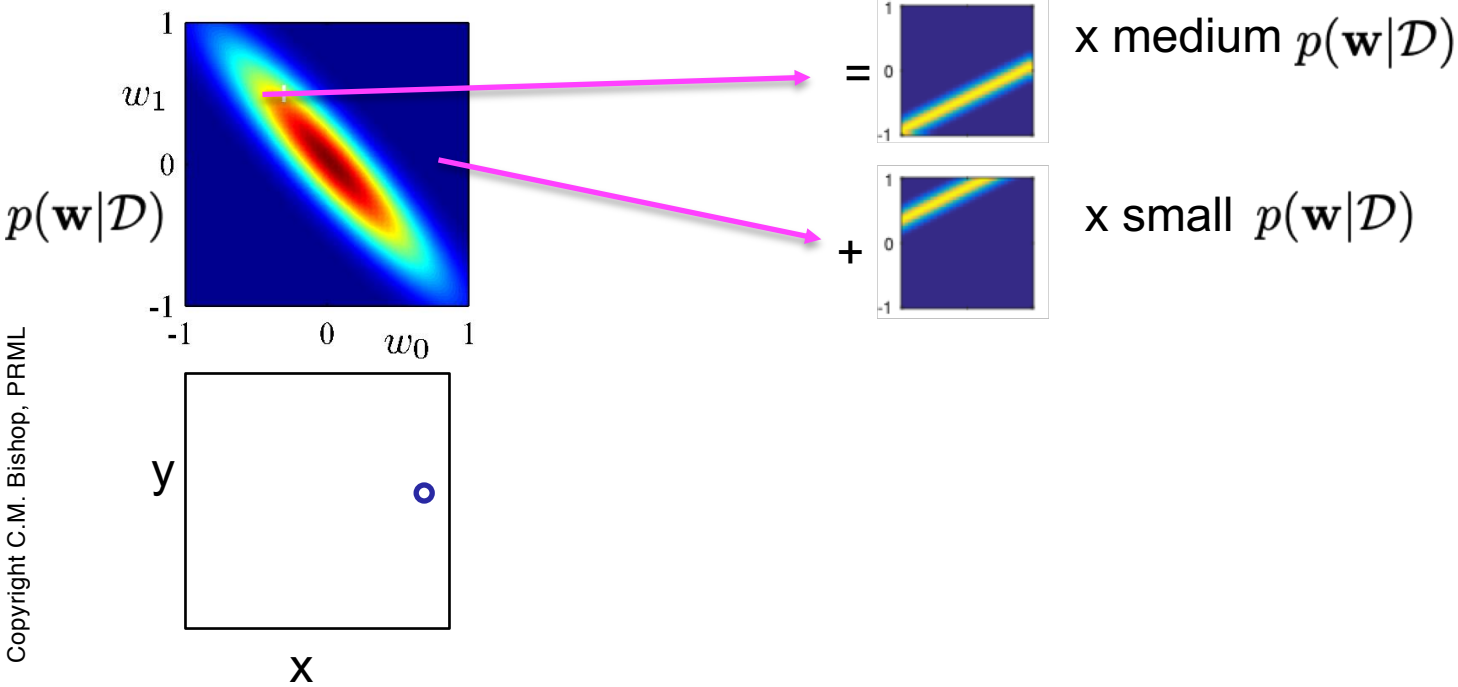
x medium $p(\mathbf{w}|\mathcal{D})$

Copyright C.M. Bishop, PRML

Step 4: prediction

- Every **w** makes a prediction, weighted by posterior

$$\int_{\mathbb{R}^N} p(\mathbf{w}|\mathcal{D})p(y^*|\mathbf{x}^*,\mathbf{w})d\mathbf{w}$$

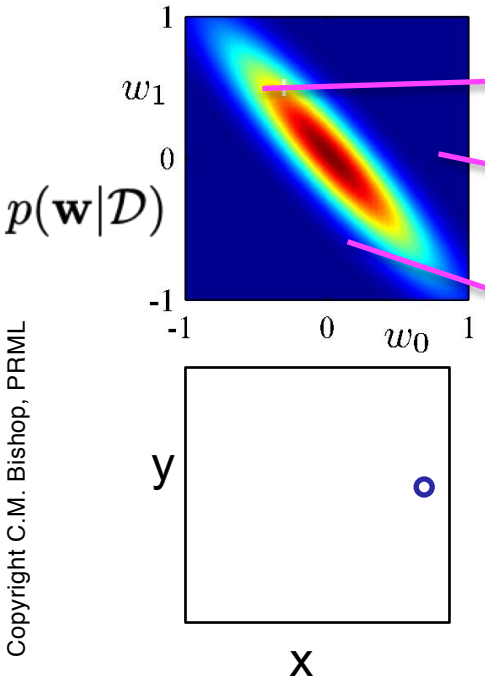


Copyright C.M. Bishop, PRML

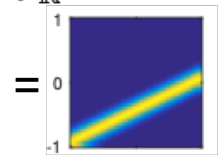
Step 4: prediction

- Every **w** makes a prediction, weighted by posterior

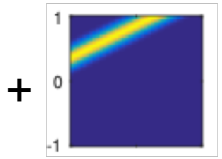
$$\int_{\mathbb{R}^N} p(\mathbf{w}|\mathcal{D})p(y^*|\mathbf{x}^*,\mathbf{w})d\mathbf{w}$$



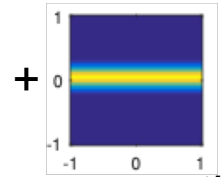
Copyright C.M. Bishop, PRML



x medium $p(\mathbf{w}|\mathcal{D})$



x small $p(\mathbf{w}|\mathcal{D})$

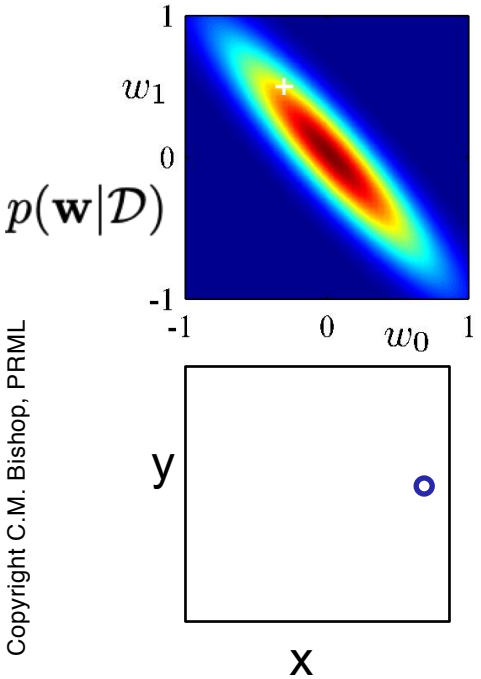


x large $p(\mathbf{w}|\mathcal{D})$

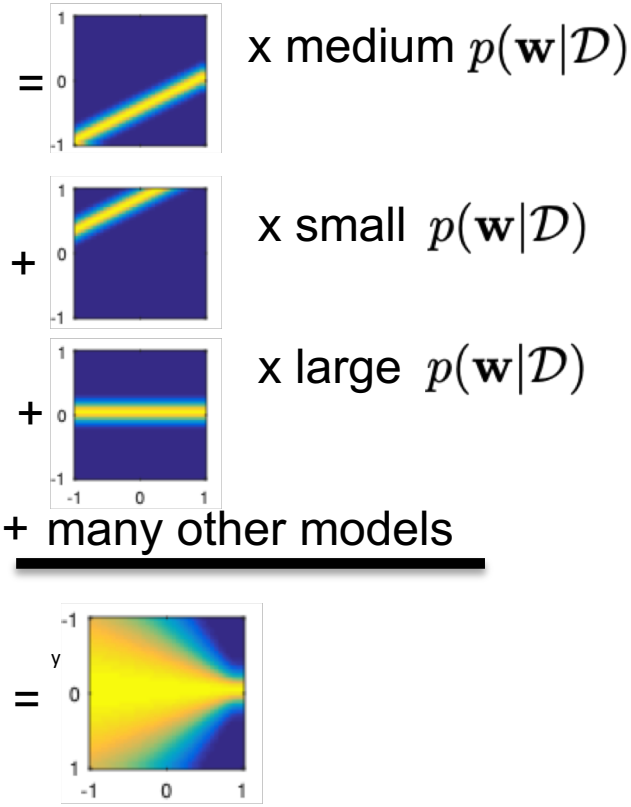
+ many other models

Step 4: prediction

- Every **w** makes a prediction, weighted by posterior



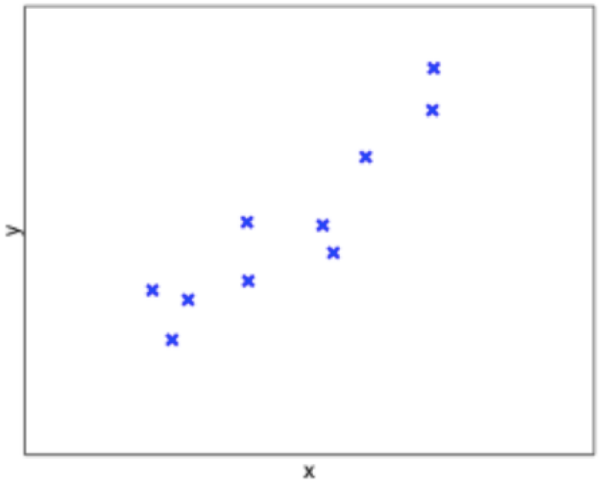
Copyright C.M. Bishop, PRML



Bayesian linear regression

- Like ordinary linear regression, can use non-linear basis

$$f_w(x) = w_0 + w_1 x + w_2 x^2$$

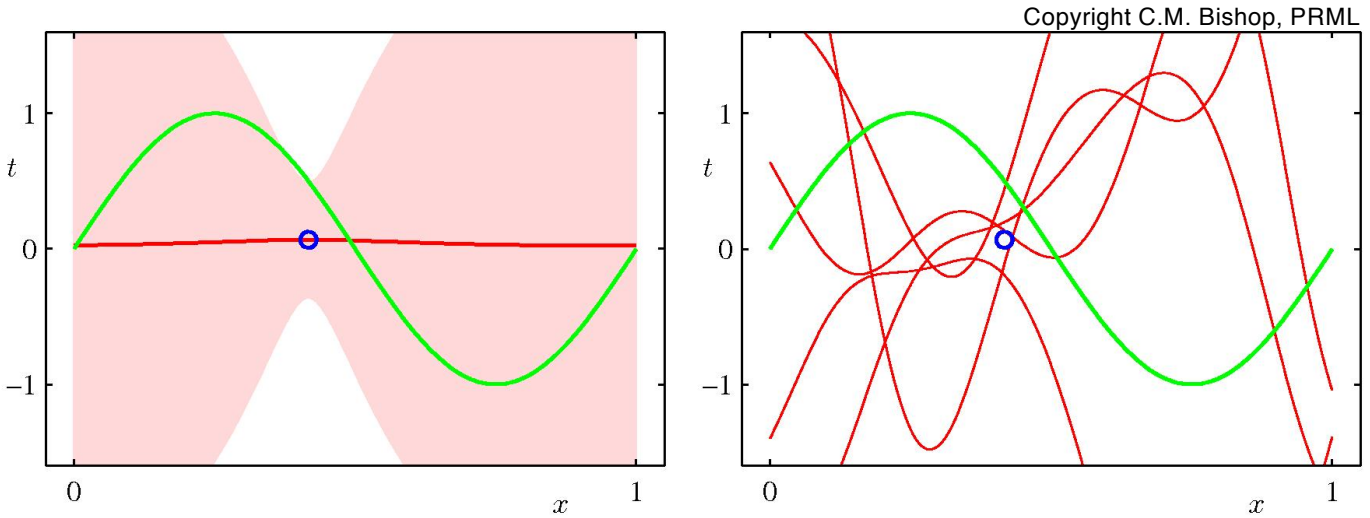


$$X = \begin{bmatrix} x^2 & x & \\ 0.75 & 0.86 & 1 \\ 0.01 & 0.09 & 1 \\ 0.73 & -0.85 & 1 \\ 0.76 & 0.87 & 1 \\ 0.19 & -0.44 & 1 \\ 0.18 & -0.43 & 1 \\ 1.22 & -1.10 & 1 \\ 0.16 & 0.40 & 1 \\ 0.93 & -0.96 & 1 \\ 0.03 & 0.17 & 1 \end{bmatrix} \quad Y = \begin{bmatrix} 2.49 \\ 0.83 \\ -0.25 \\ 3.10 \\ 0.87 \\ 0.02 \\ -0.12 \\ 1.81 \\ -0.83 \\ 0.43 \end{bmatrix}$$

Lecture 3, linear regression

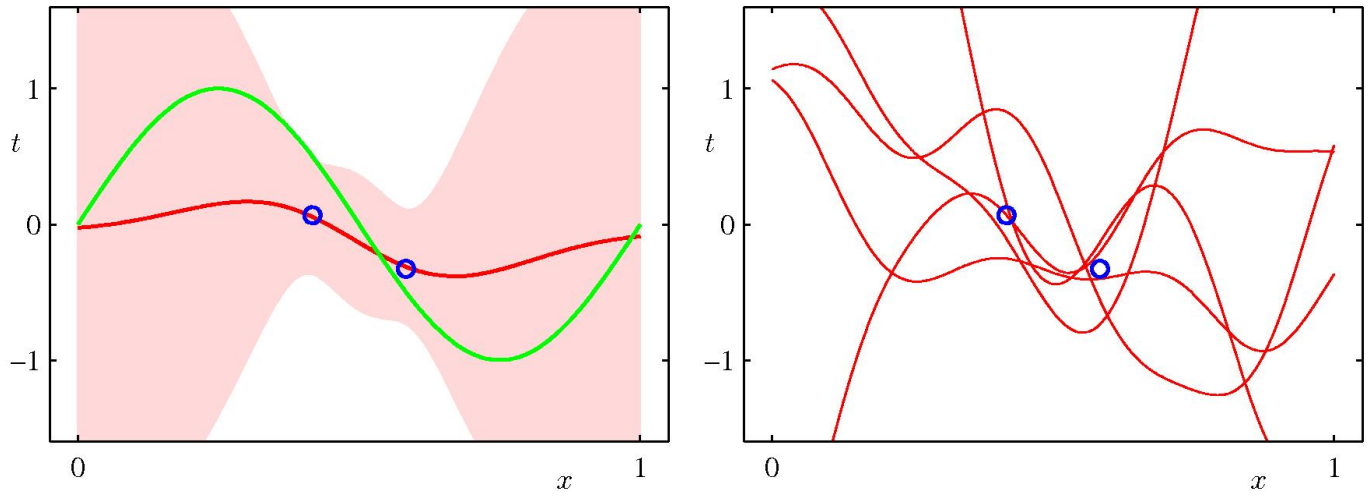
Example: Polynomial bases

- Example: Bayesian linear regression with polynomial bases



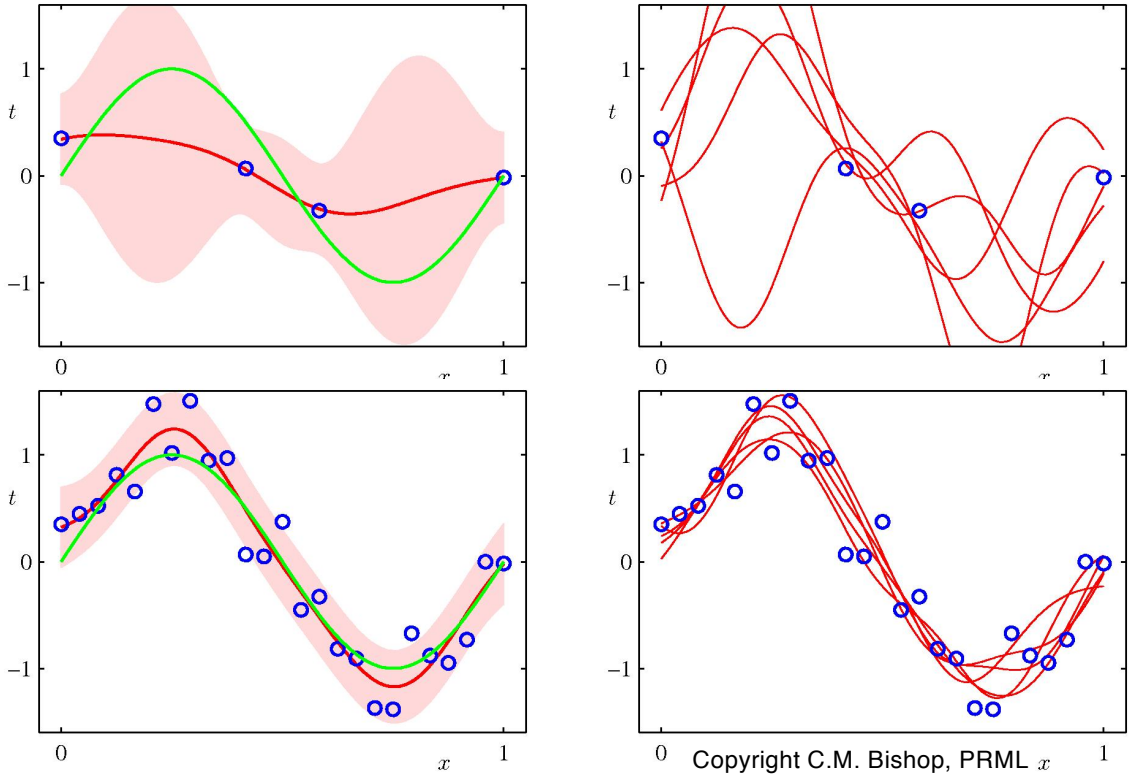
- Green line: true function. Blue circles: data points.
  Red line: MAP prediction. Shaded red: posterior predictive distribution.

Example: Polynomial bases



Copyright C.M. Bishop, PRML

Example: Polynomial bases



Application: black-box optimization

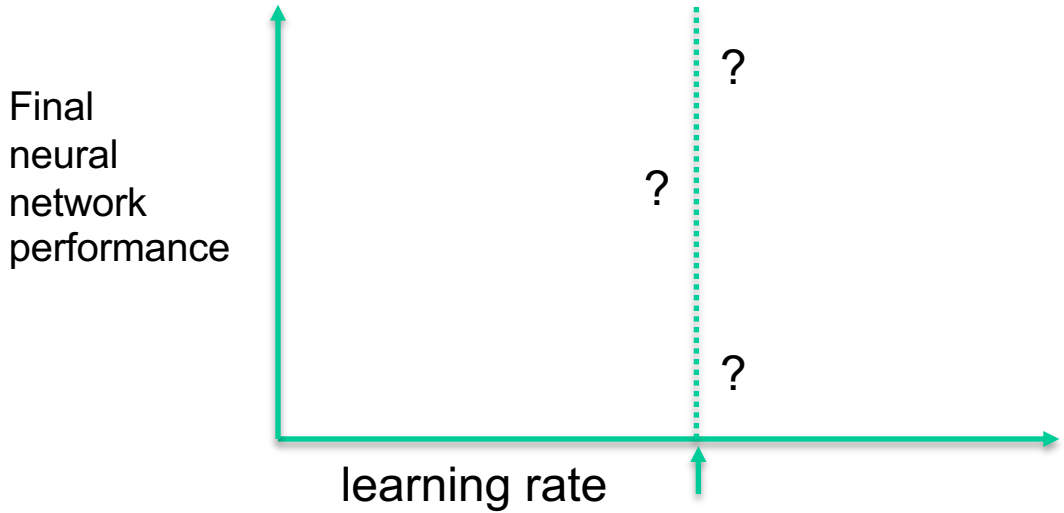
- Problem: find value x for which function $f(x)$ is maximized

- Constraints:
  - $f(x)$ is a 'black box' function: we only know the value $f(x)$ for small set of points x that we evaluate
  - Function may be highly non-convex.
  - Evaluating $f(x)$ is relatively expensive
  - Derivatives might not be known

- Example: finding the hyperparameters of a neural network
- How can we approach this problem?

Black-box optimization

- Problem: find value x for which function f(x) is maximal
- Example of black box function



Black-box optimization

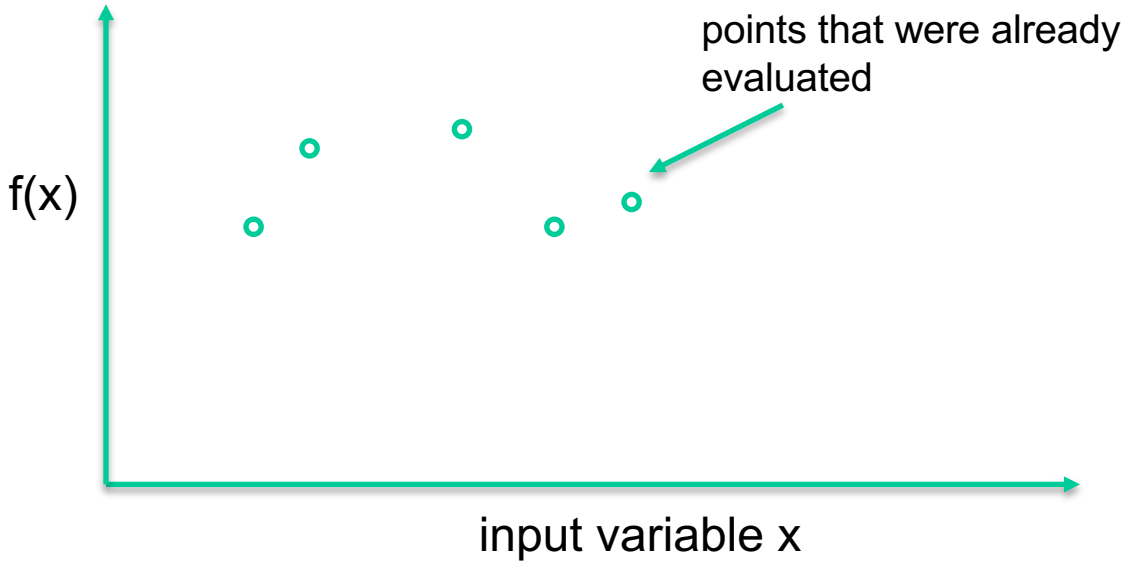
- So far, we have mainly done optimization via gradient decent.

- But gradient descent requires an estimate of the gradient
  - Might need many function evaluations (costly)
  - Might not know derivatives

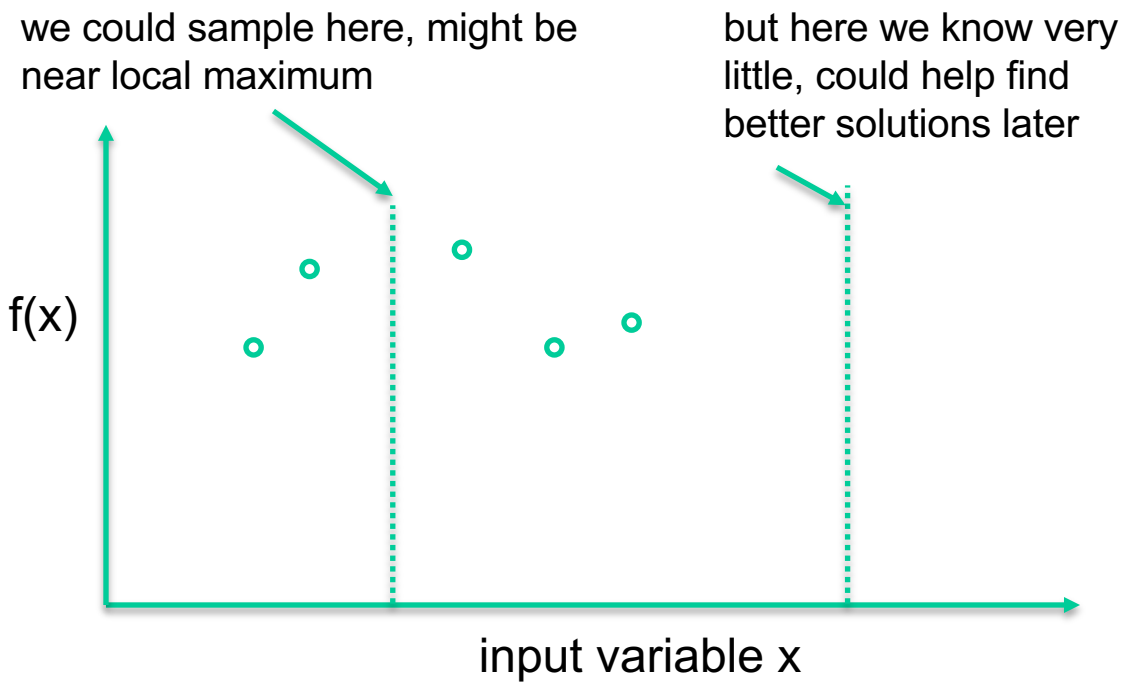
- What can we do in this "black box" situation?

Black-box optimization

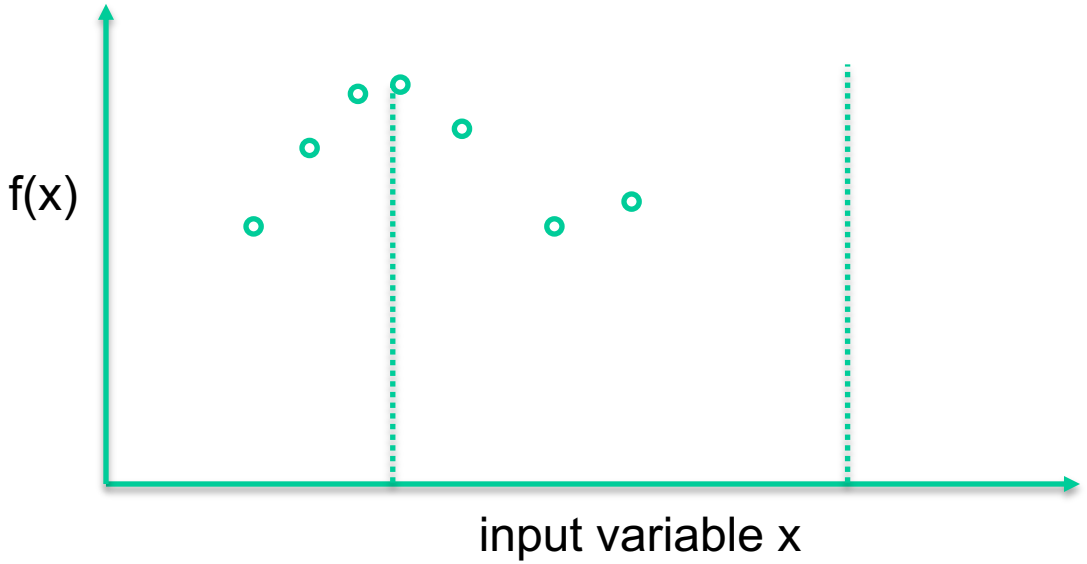
- Where to sample next, if we have a budget for, say, 10 samples?



Black-box optimization



Black-box optimization

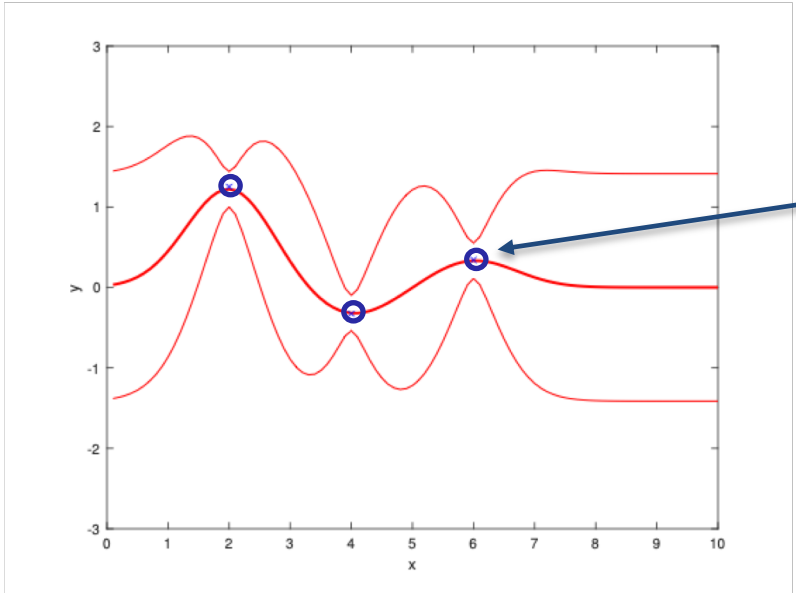


Bayesian optimization

- Idea: to make a good decision we should *imagine what the whole function should look like*
- It seems important to take into account how certain we are for various input values $\mathbf{x}$
- Bayesian linear regression might do the job here!
- This implies Bayesian point of view: Bayesian optimization (a method to do black-box optimization)

Bayesian optimisation

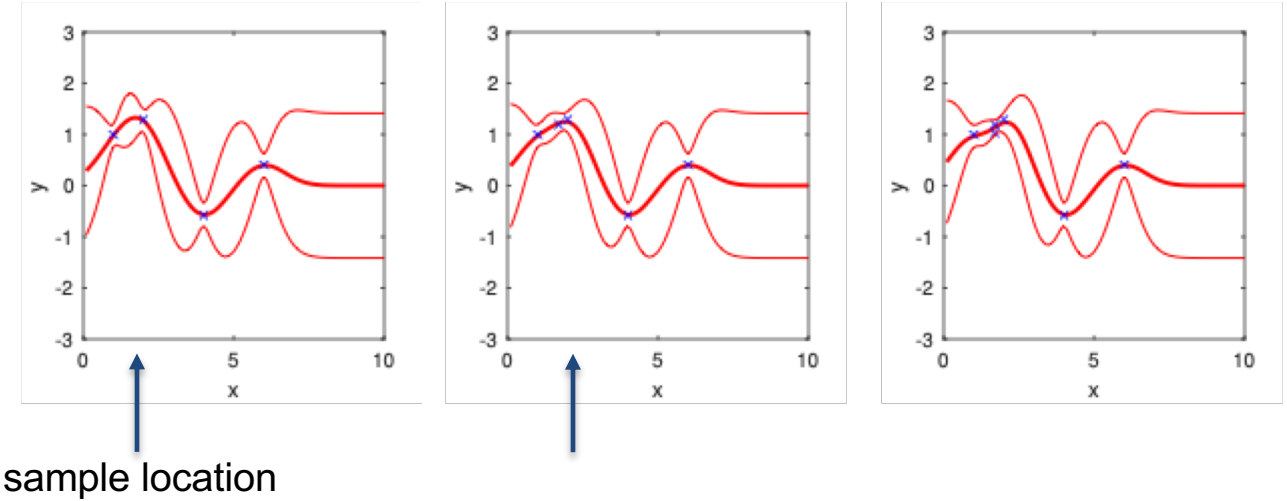
- Bayesian posterior over function



- Where to sample next?

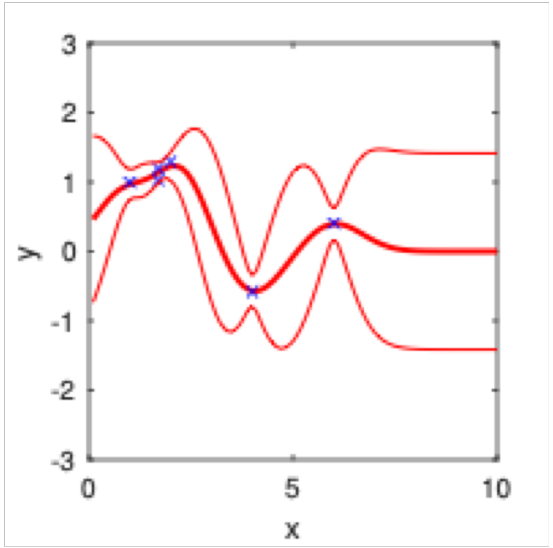
Bayesian optimisation

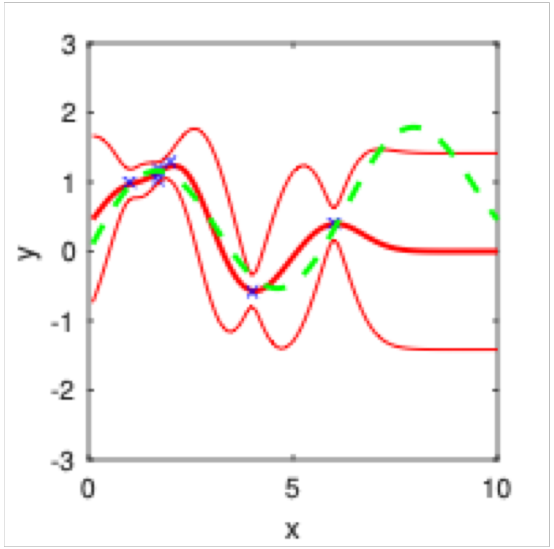
- Where to sample next?
- What happens if we simply sample where mean is highest?



Bayesian optimisation

- We might miss the real maximum!!!





Bayesian optimisation

Two objectives:

- Exploitation: sample where we think high values are.
  - If we know the samples will be low, it does not make sense to sample there
  - Maybe: sample highest mean?

- Exploration: If we always sample where we think the highest value is, we might miss other values
  - Maybe: sample where uncertainty is highest?

Bayesian optimisation

- Several strategies exist for combining these two objectives
- Can give ‘score’ to possible examples using **acquisition function**

Bayesian optimisation

- Several strategies exist for combining these two objectives
- Can give 'score' to possible examples using **acquisition function**
- Very straightforward method: upper confidence bound (UCB)

$$a_{\text{UCB}}(\mathbf{x}^*; \mathcal{D}) = \mu(\mathbf{x}^*; \mathcal{D}) + \kappa\sigma(\mathbf{x}^*; \mathcal{D})$$

predicted mean given data so far

trade-off parameter

predicted standard deviation given data so far

- Acquisition functions gives a 'score' to each sample point
- UCB has good theoretical properties

Bayesian optimisation

- Pros
  - Attempt at global optimization
  - Need relatively few samples to get close to optimum
  - Software packages available

- Cons
  - Computational expensive
  - Sensitive to choice of model
  - Only works well with few input (up to ~10 dimensions)

What you should know

- Bayesian terminology (prior, posterior, likelihood, etc.)
- Conjugate priors, what they mean.
- Bayesian linear regression and its properties
- When and why to use Bayesian methods
- Core concepts behind Bayesian optimization