# Bayesian Machine Learning

**Roadmap:**

— What is Machine Learning?
— Why/when Bayesian ML?
— Bayesian Interpolation
— Model complexity

**Iain Murray**

School of Informatics, University of Edinburgh

# What is Machine Learning?

(Depends who you ask)

***"Getting computers to use data to perform better at a task."***

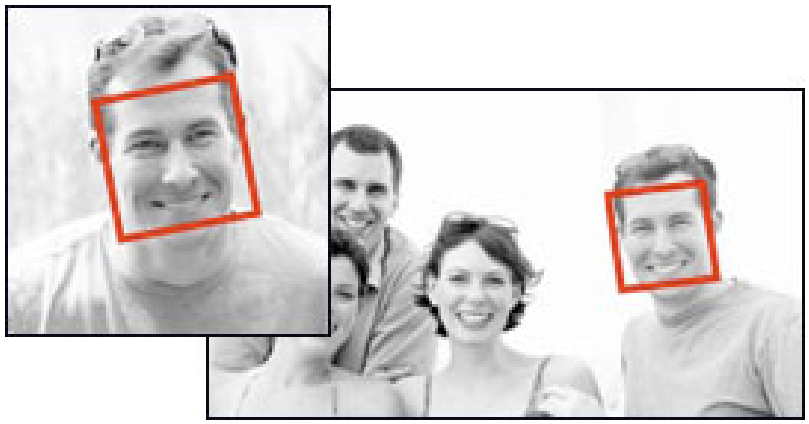
**Applications:**

— Pattern Recognition, e.g., classification
— Inferring processes underlying data
— Controllers
...

# Face detection

How would you detect a face?

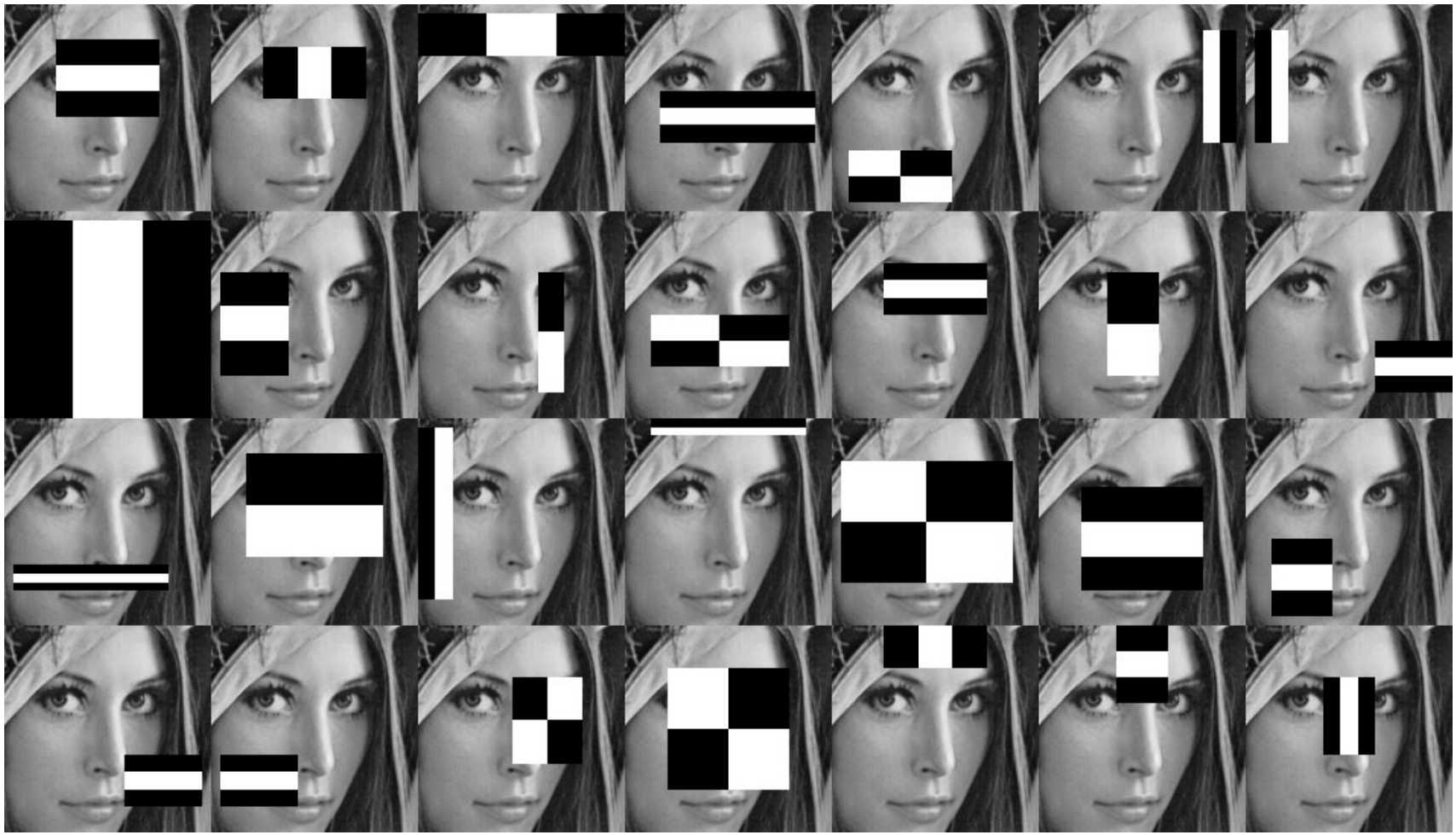
(R. Vaillant, C. Monrocq and Y. LeCun, 1994)



How does album software tag your friends?

http://demo.pittpatt.com/

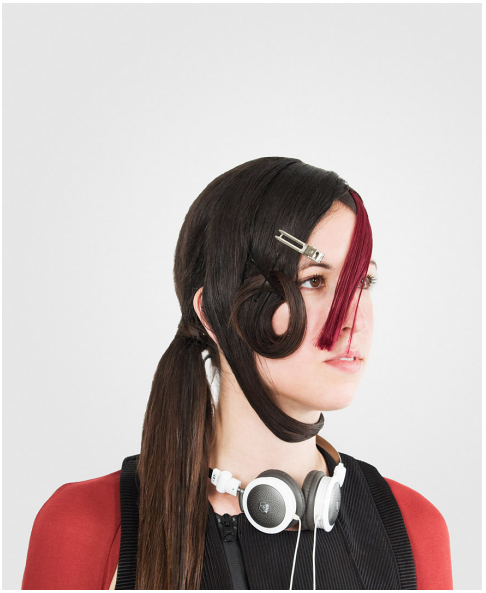
# Face detection



Taken from: http://v10.ahprojects.com/art/cv-dazzle

# How do humans do it?



Taken from: http://v10.ahprojects.com/art/cv-dazzle

# Response surface optimization

Consult on making a new:
concrete, weld, . . . widget?

(antmoose on Flickr, cc-by-2.0)

(Bhadeshia et al.)

# Bayesian Machine learning (examples)

**Can be uncertain, even with lots of data:**

Bayesian neural networks (MacKay 1995; Neal, 1996)

Bayesian Sets (Ghahramani and Heller, 2006)

TrueSkill (Herbrich, Minka and Graepel, 2006)

Bayesian Matrix Factorization (Salakhutdinov and Mnih, 2008)

...

Often only indirect or limited data about what you're interested in at any given time.

# Overview

**Machine learning:** *fit a bunch of numbers from data*

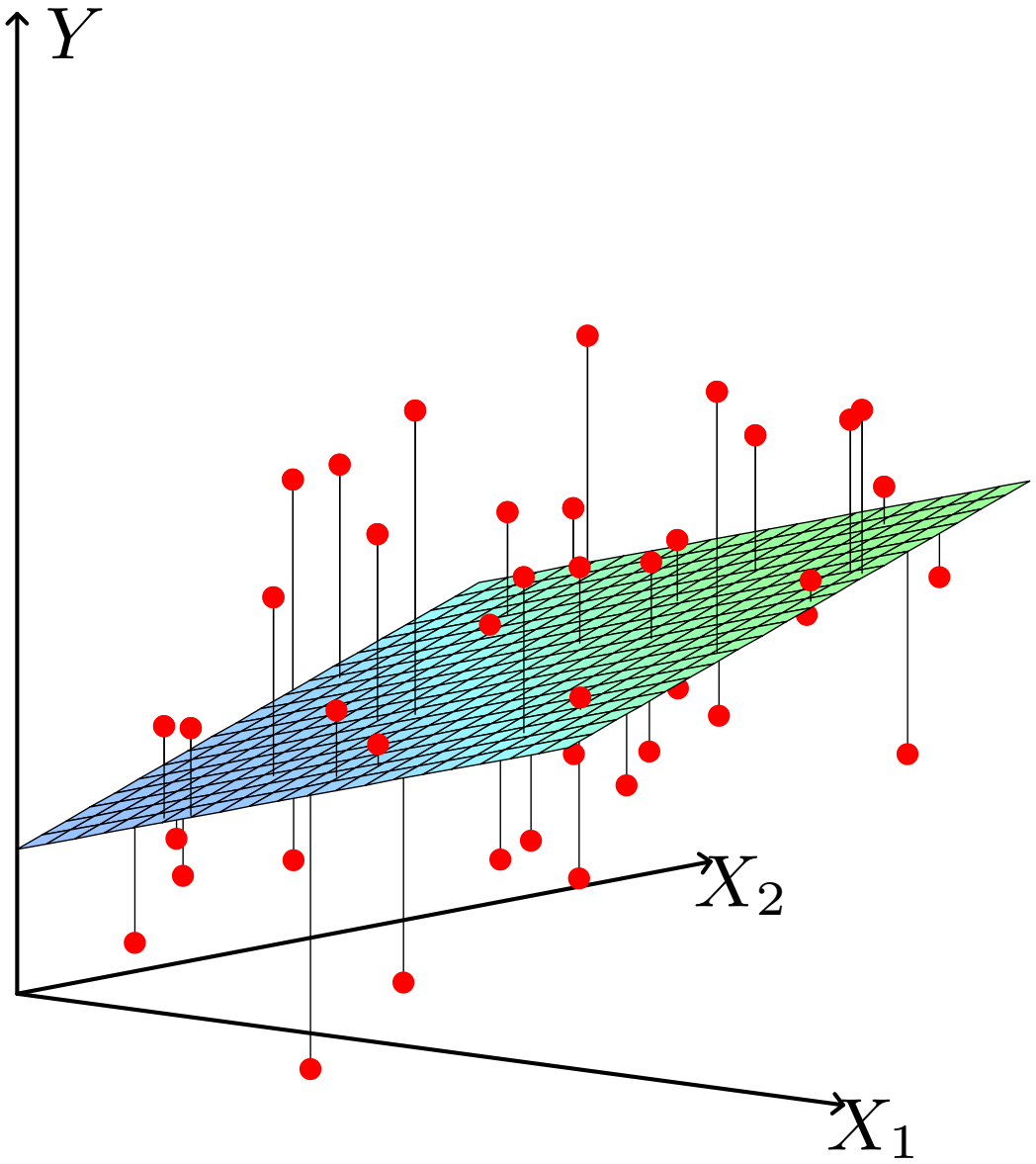
**Bayesian:** *optimal inference with limited information*

How much human tweaking is required?

Can our machines have insight?

**This lecture:** Bayesian interpolation

# Linear regression




Find linear function

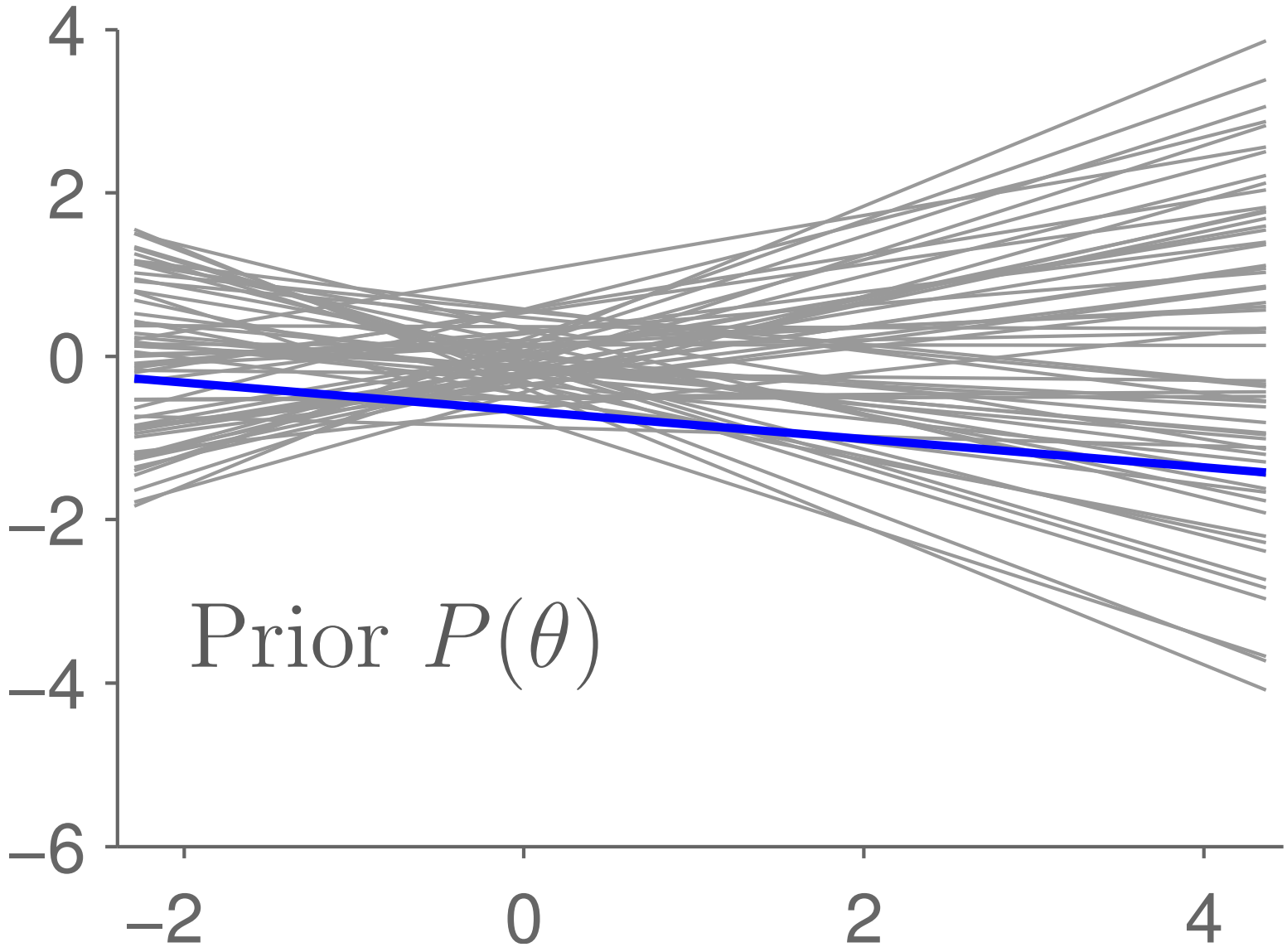
$$\hat{\mathbf{y}} = X\mathbf{w}$$

that minimizes sum of squared residuals from $\mathbf{y}$.

Matlab/Octave:

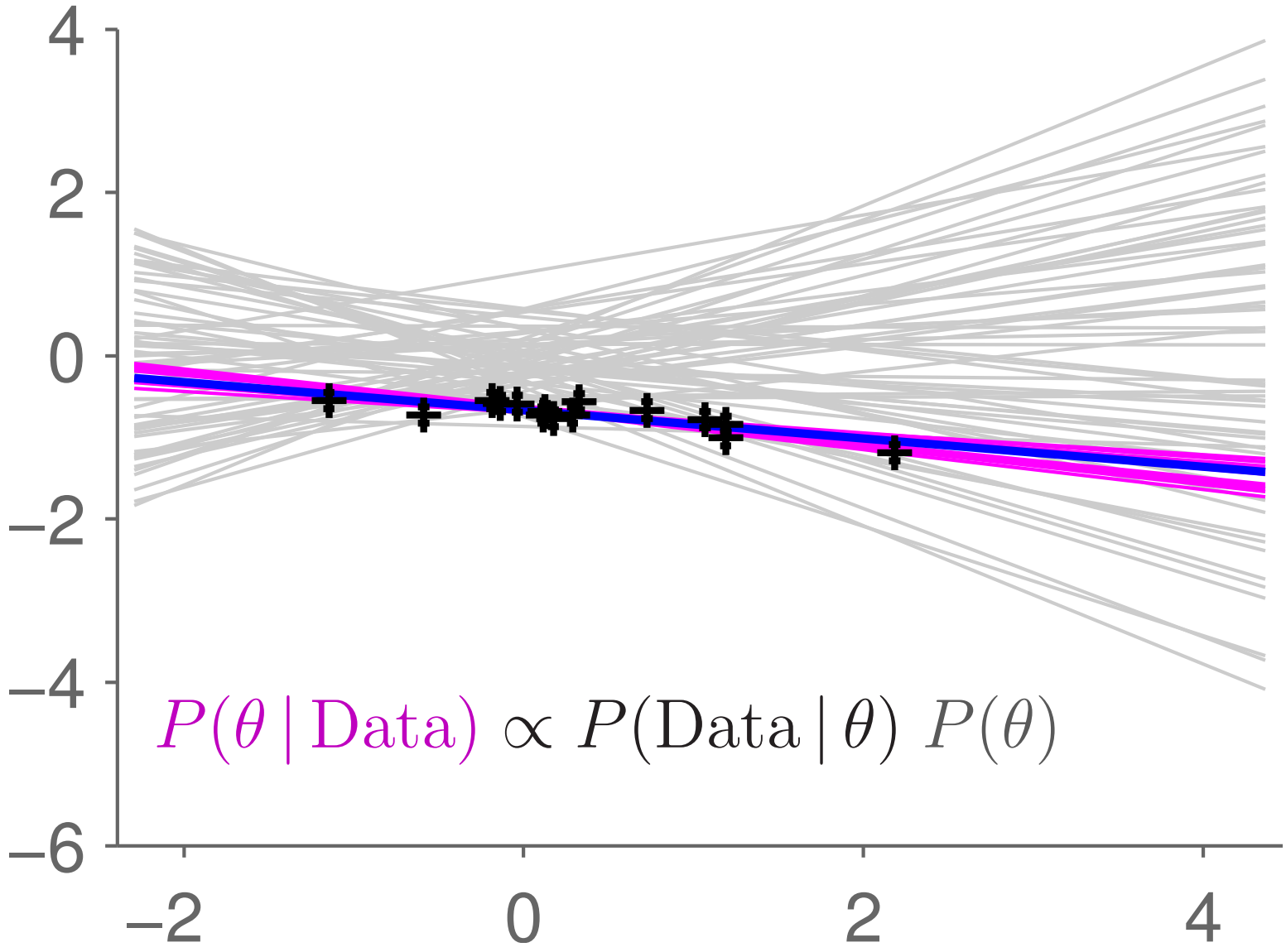
```
ww = X \ yy
```

# Linear Regression: Prior



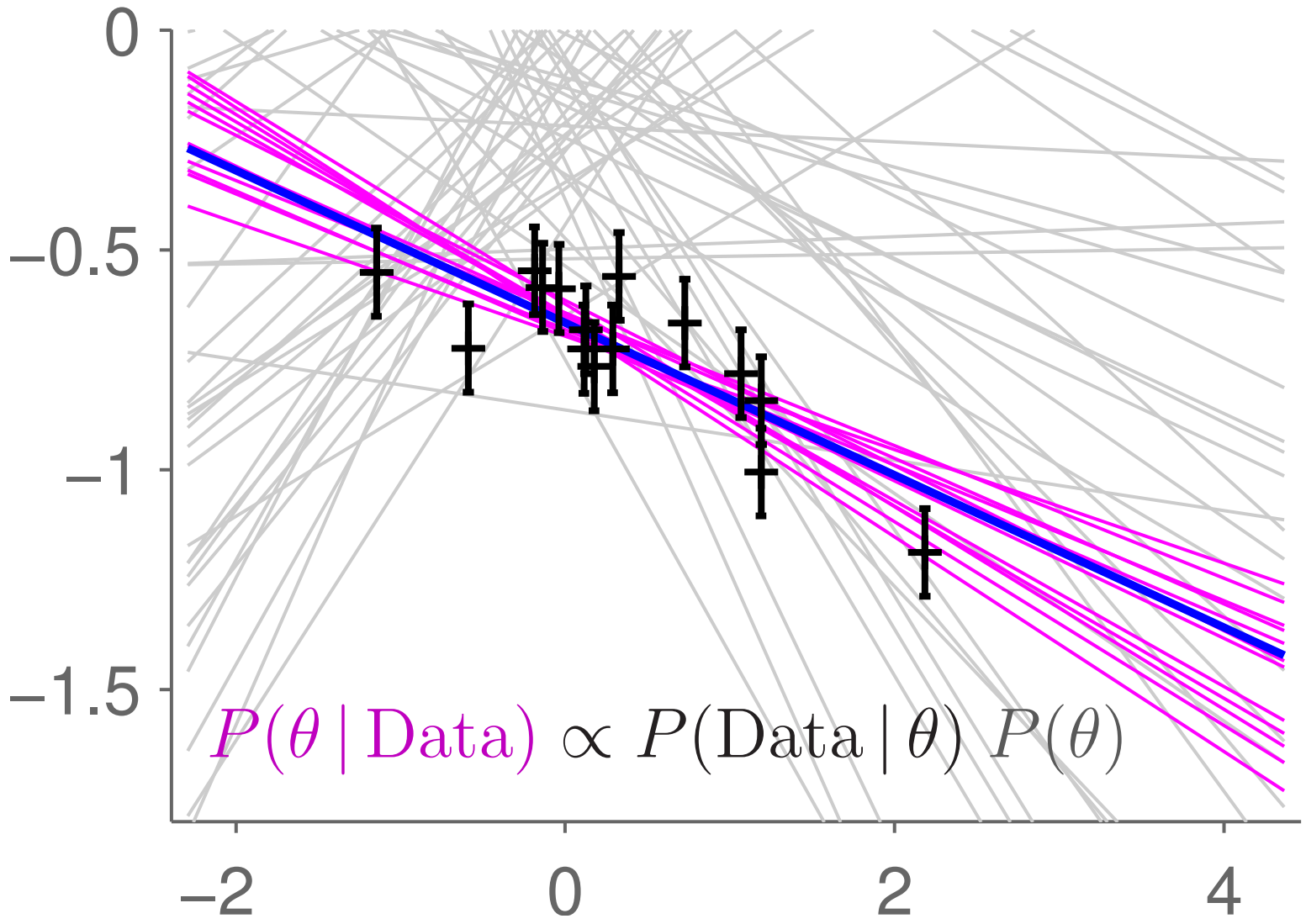

Input $\rightarrow$ output mappings considered plausible before seeing data.

# Linear Regression: Posterior



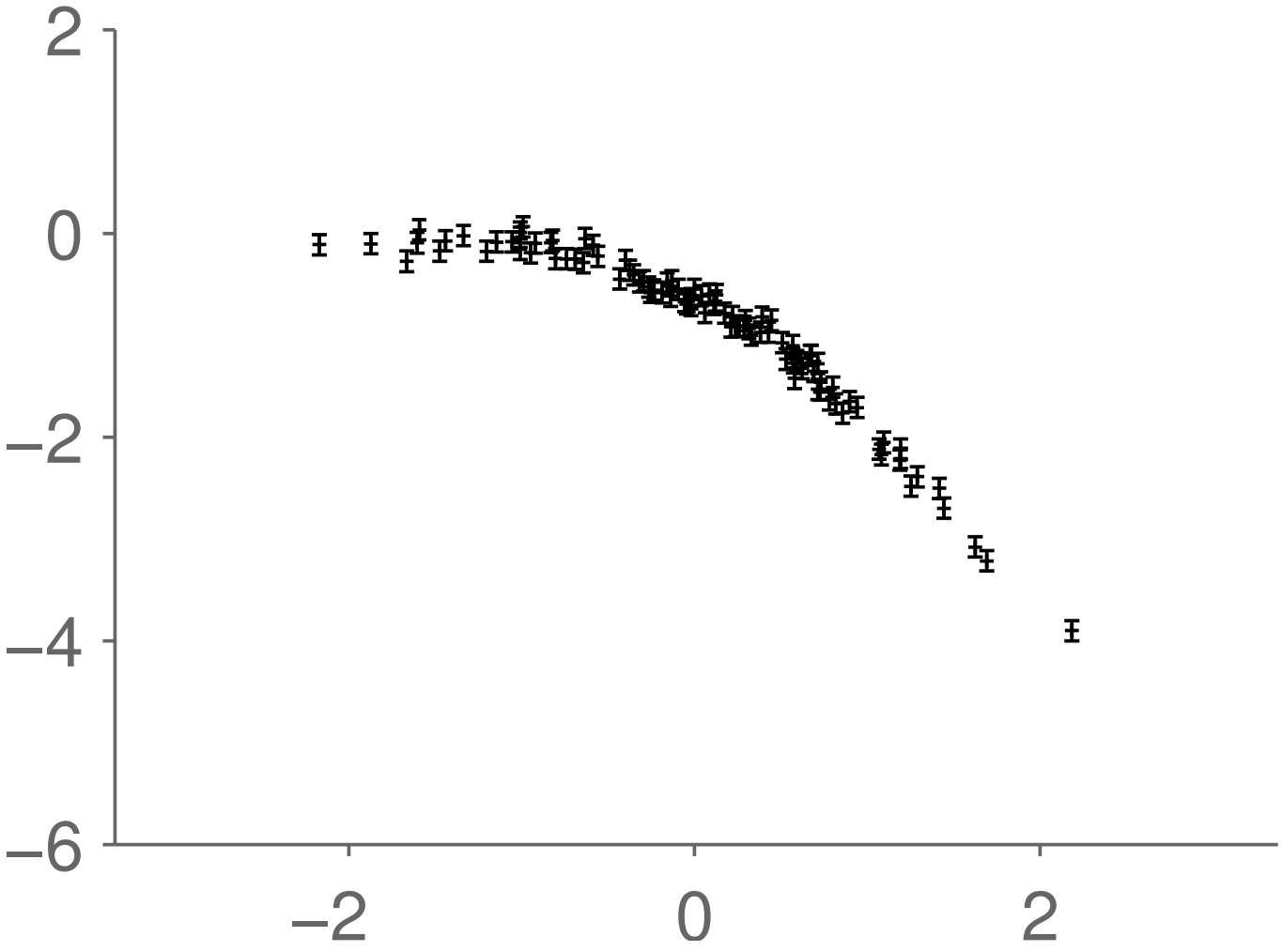

Posterior much more compact than prior.

# Linear Regression: Posterior




Draws from posterior. Non-linear error envelope. Possible explanations linear.

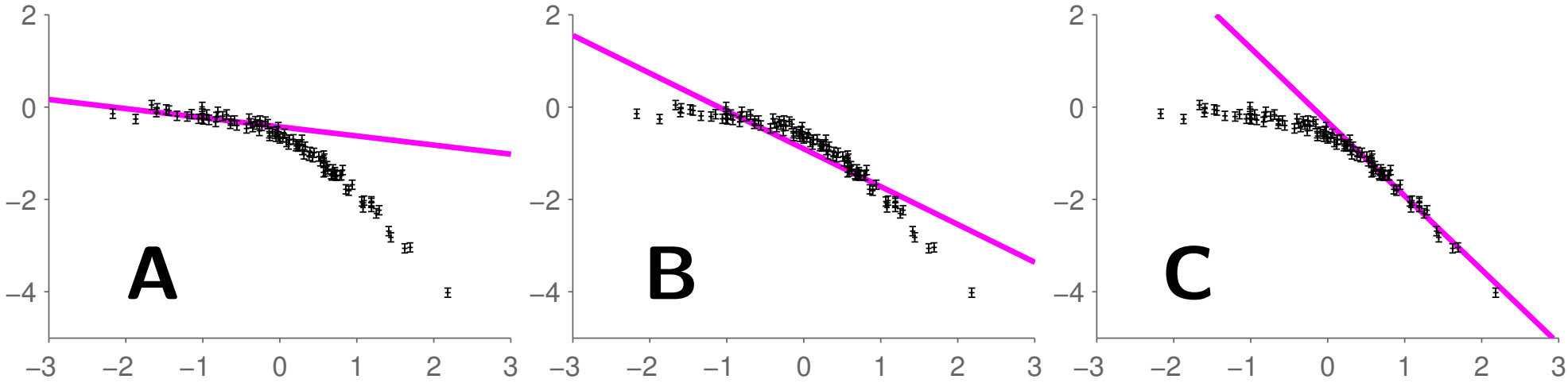
# Model mismatch




What will Bayesian linear regression do?

# Quiz

*Given a (wrong) linear assumption*, which explanations are typical of the posterior distribution?

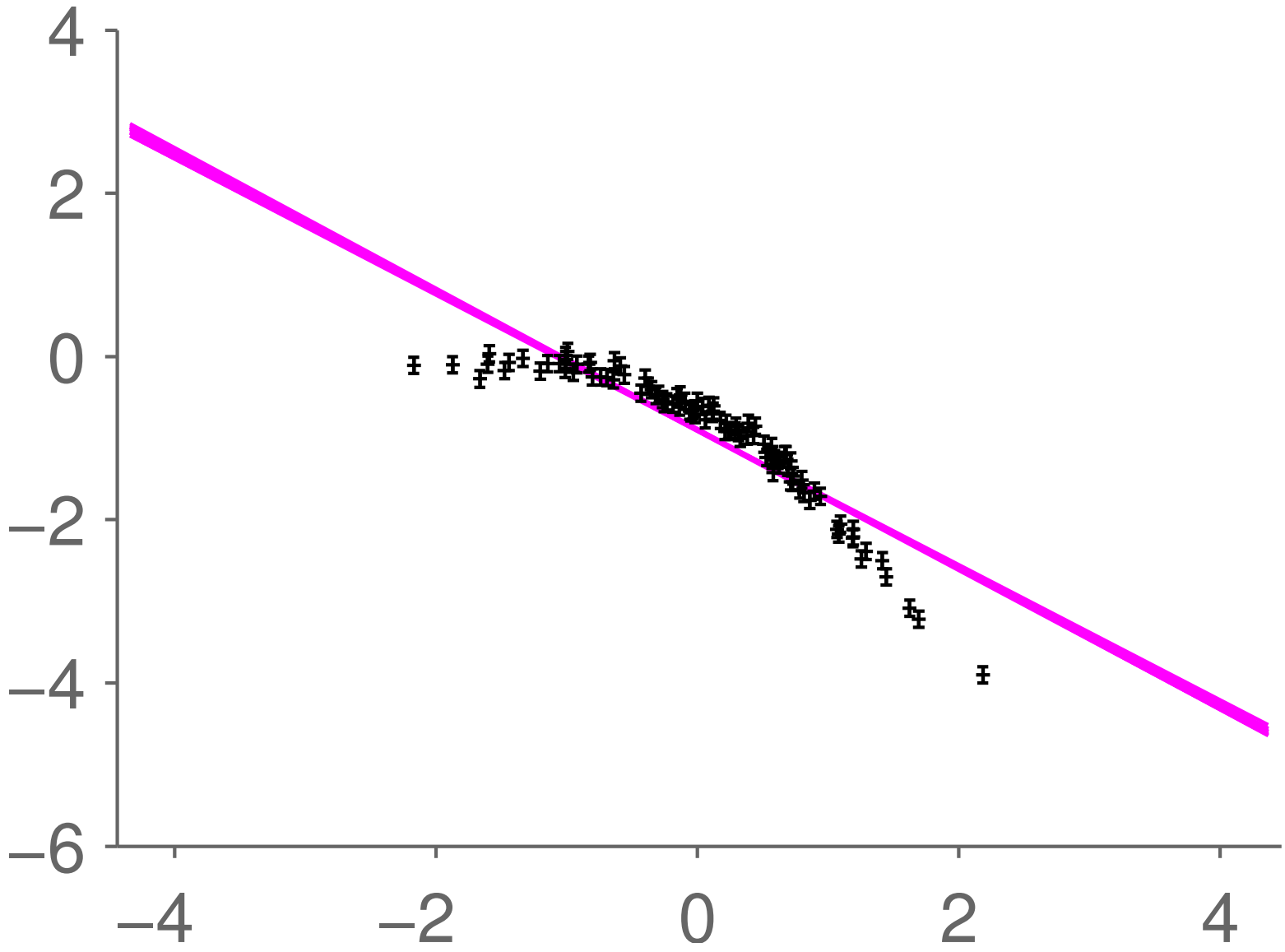



**D** All of the above

**E** None of the above

**Z** Not sure

# ‘Underfitting’




Posterior *very* certain despite blatant misfit. Prior ruled out truth.

# Linear regression

Have code for fitting $\mathbf{y} \approx X\mathbf{w}$

$$\mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix} \qquad X = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_D^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_D^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \cdots & x_D^{(N)} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \vdots \\ \mathbf{x}^{(N)} \end{bmatrix}$$

Jargon: the $N \times D$ matrix $X$ is called the *design matrix*.

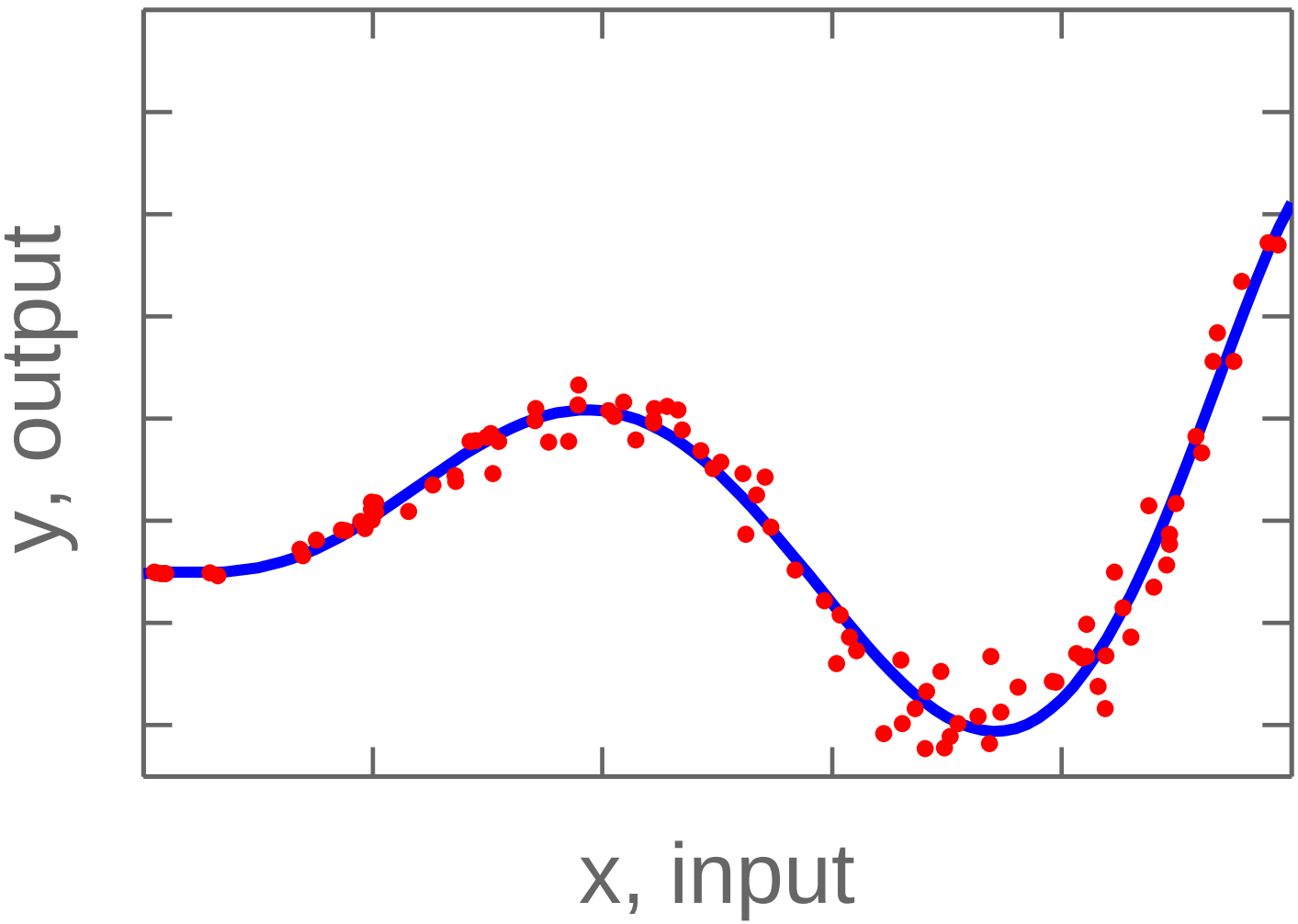
Statisticians often use $p$ instead of $D$ for the number of features (AKA *covariates*).

# Linear regression (with features)

$$y \approx w_1 + w_2\, x + w_3\, x^2 = \phi(x) \cdot \mathbf{w} \quad \text{or} \quad \mathbf{y} \approx X\mathbf{w}\text{:}$$

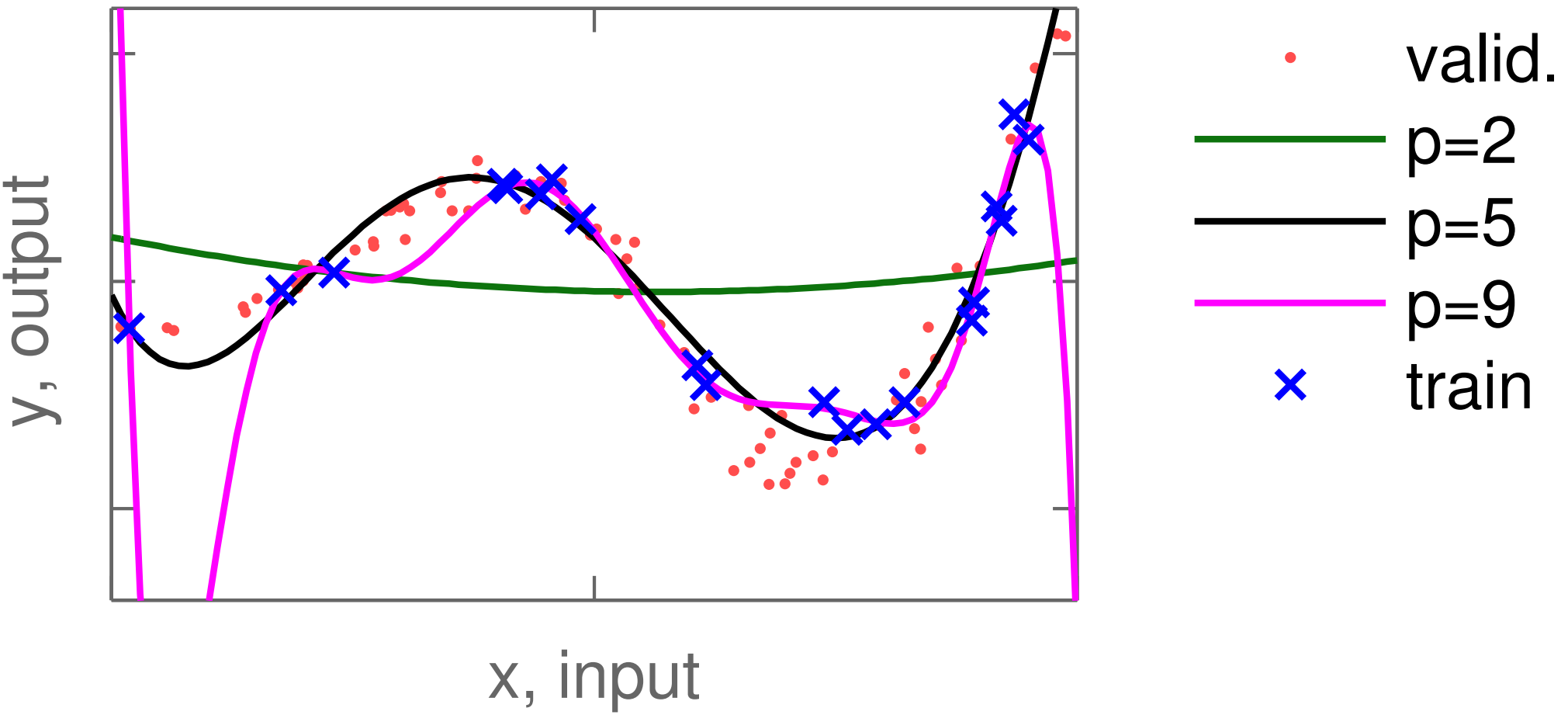
$$\mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix} \qquad X = \begin{bmatrix} 1 & x^{(1)} & (x^{(1)})^2 \\ 1 & x^{(2)} & (x^{(2)})^2 \\ \vdots & \vdots & \vdots \\ 1 & x^{(N)} & (x^{(N)})^2 \end{bmatrix} = \begin{bmatrix} \phi(x^{(1)}) \\ \phi(x^{(2)}) \\ \vdots \\ \phi(x^{(N)}) \end{bmatrix}$$

# Linear regression (with features)



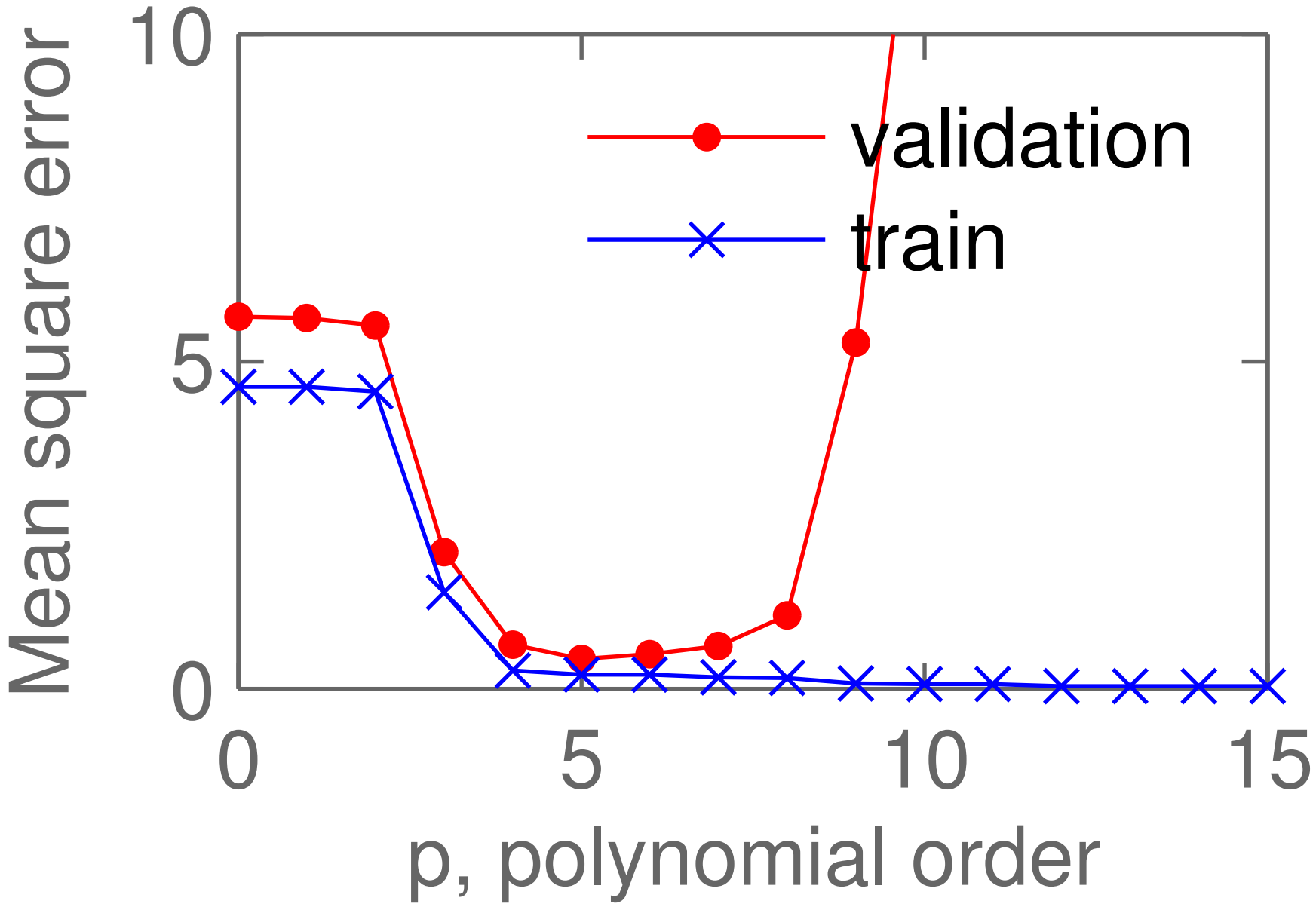

```
X = [ones(N, 1), xx, xx.^2, xx.^3, xx.^4, xx.^5, xx.^6];
Xnew = [ones(N, 1), xnew, xnew.^2, xnew.^3, xnew.^4, xnew.^5, xnew.^6]
ww = X \ yy;
ynew = Xnew * ww;  plot(xnew, ynew, '-b');
```

# Overfitting

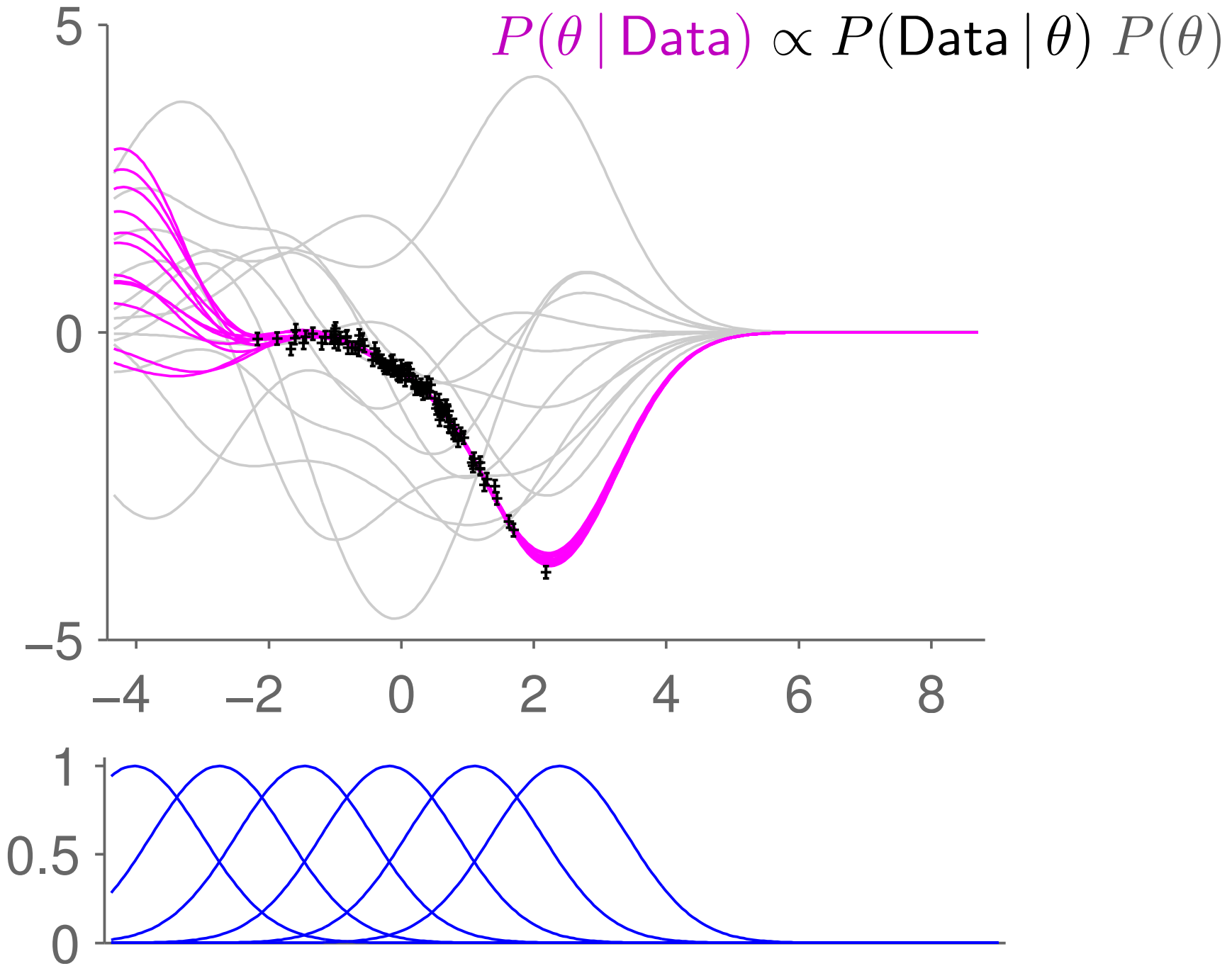



```
X = [x.^0, x.^1, x.^2, ..., x.^p];
```

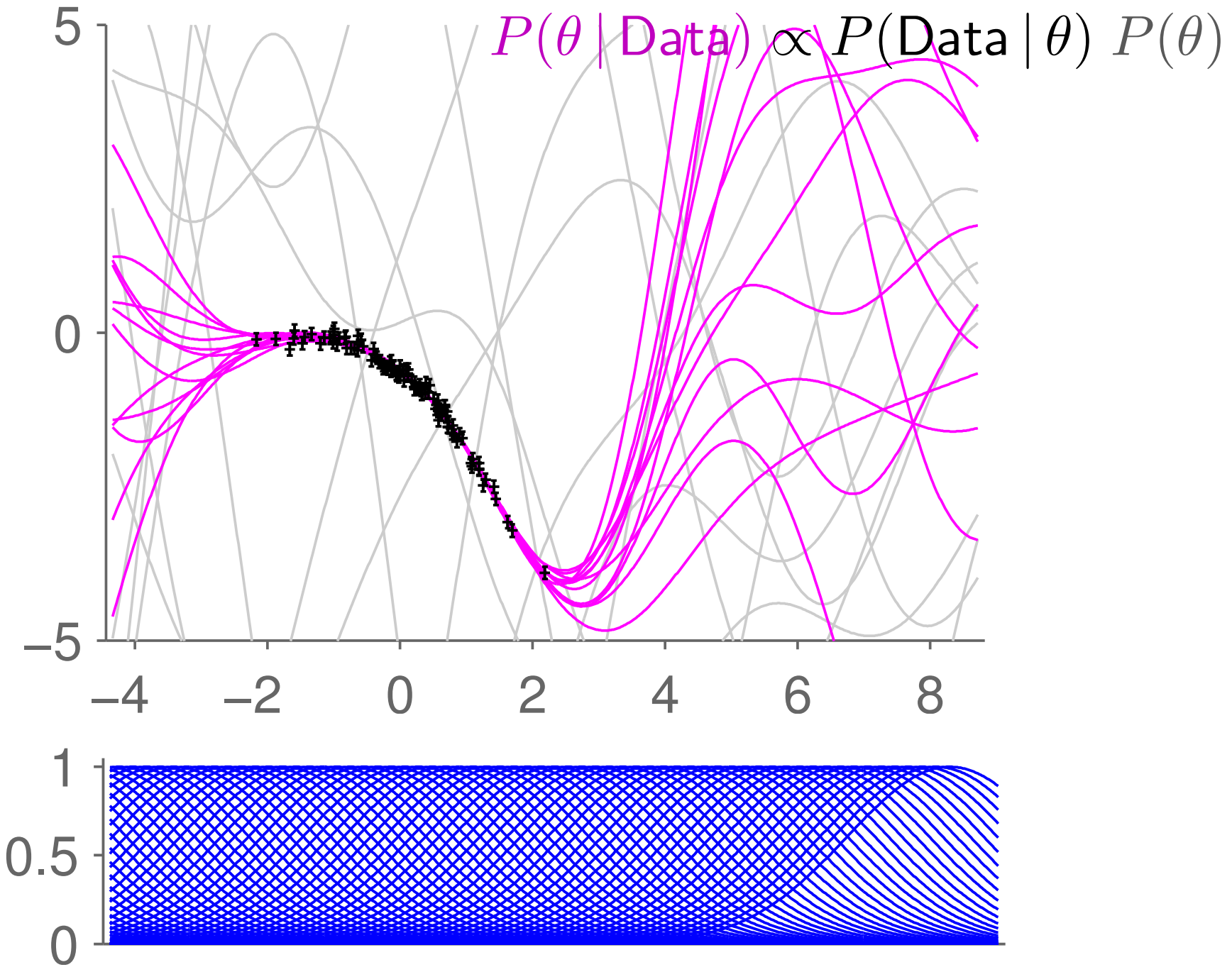
# Learning curves



# Bayesian linear regression (with RBF features)

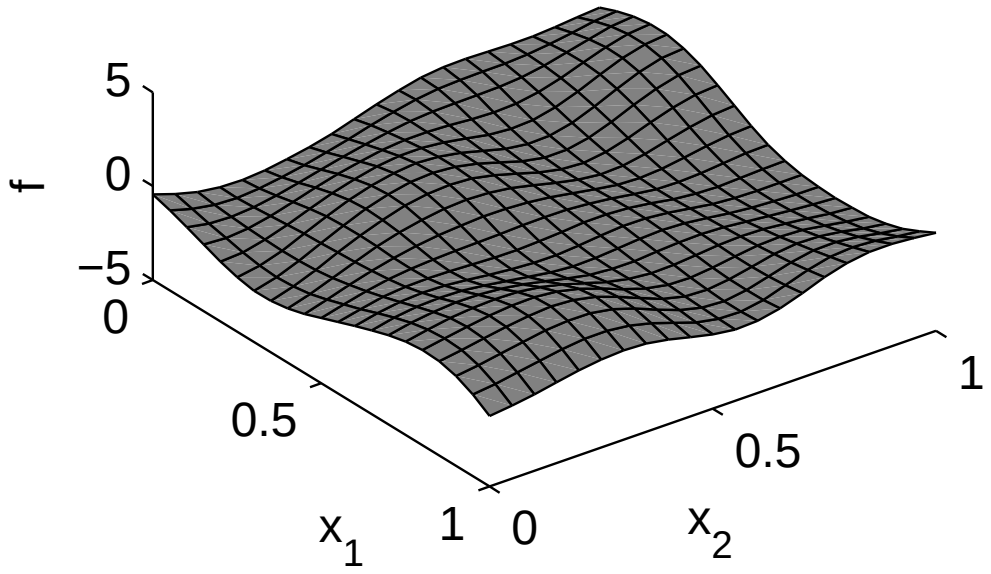


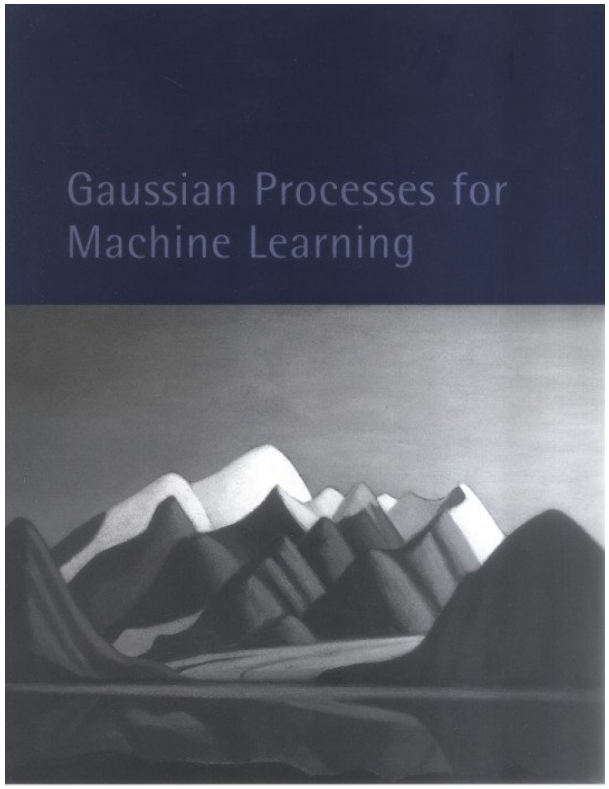
# Bayesian linear regression (with RBF features)



# Gaussian Processes

## Put basis functions *everywhere*




Carl Edward Rasmussen and Christopher K. I. Williams

`http://www.gaussianprocess.org/gpml/`

## Monte Carlo inference: (for non-Gaussian observations)

`http://homepages.inf.ed.ac.uk/imurray2/pub/10ess/`
`http://homepages.inf.ed.ac.uk/imurray2/pub/10hypers/`

# Overview

Bad to rule out truth in prior

Can use large models with little data

Q. What about Occam's razor?

# Bayesian model comparison

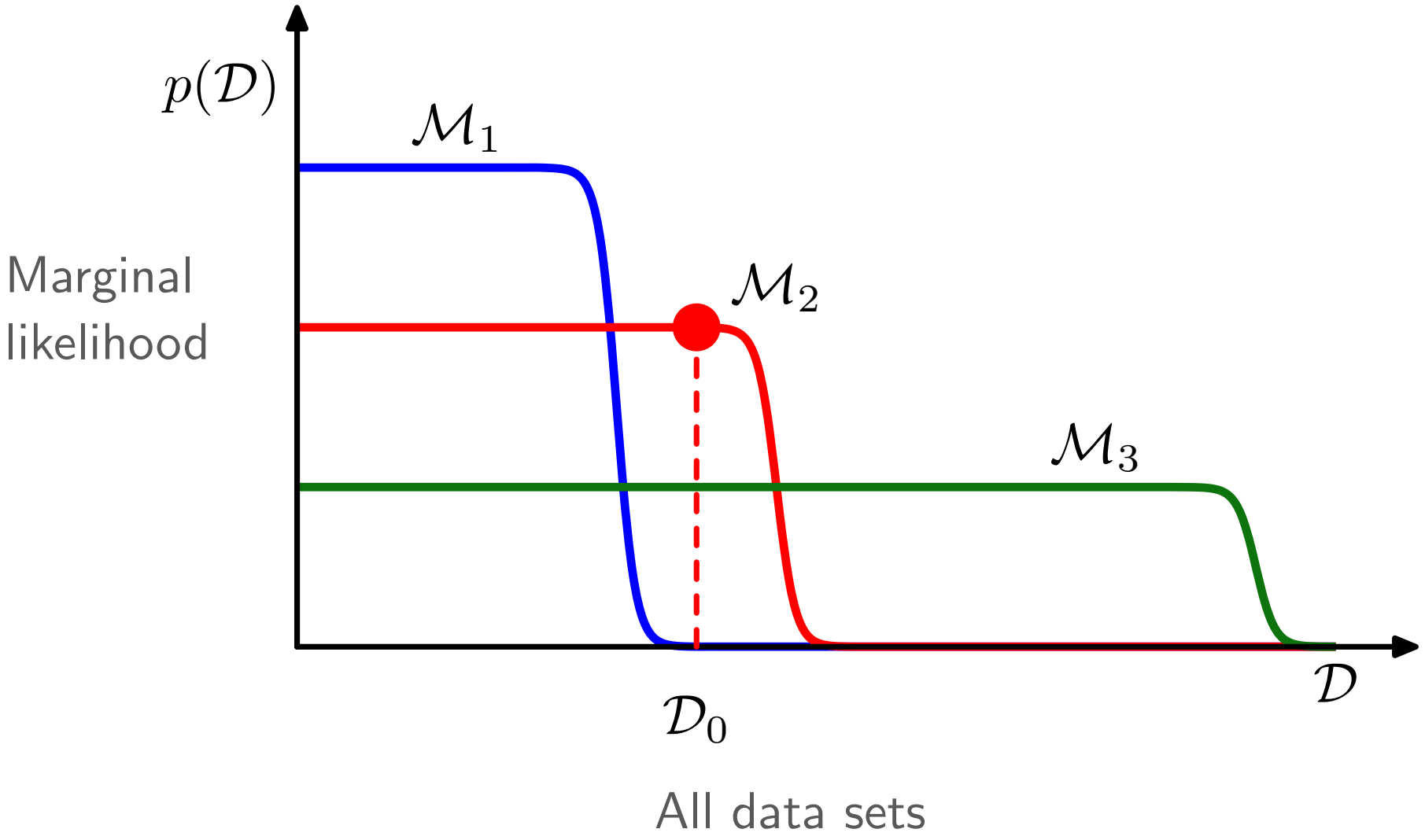
**Parameter posterior:**

$$P(\theta \,|\, \mathcal{D}, \mathcal{M}) \;=\; \frac{P(\mathcal{D} \,|\, \theta, \mathcal{M})\; P(\theta \,|\, \mathcal{M})}{P(\mathcal{D} \,|\, \mathcal{M})}$$

**Model posterior:**

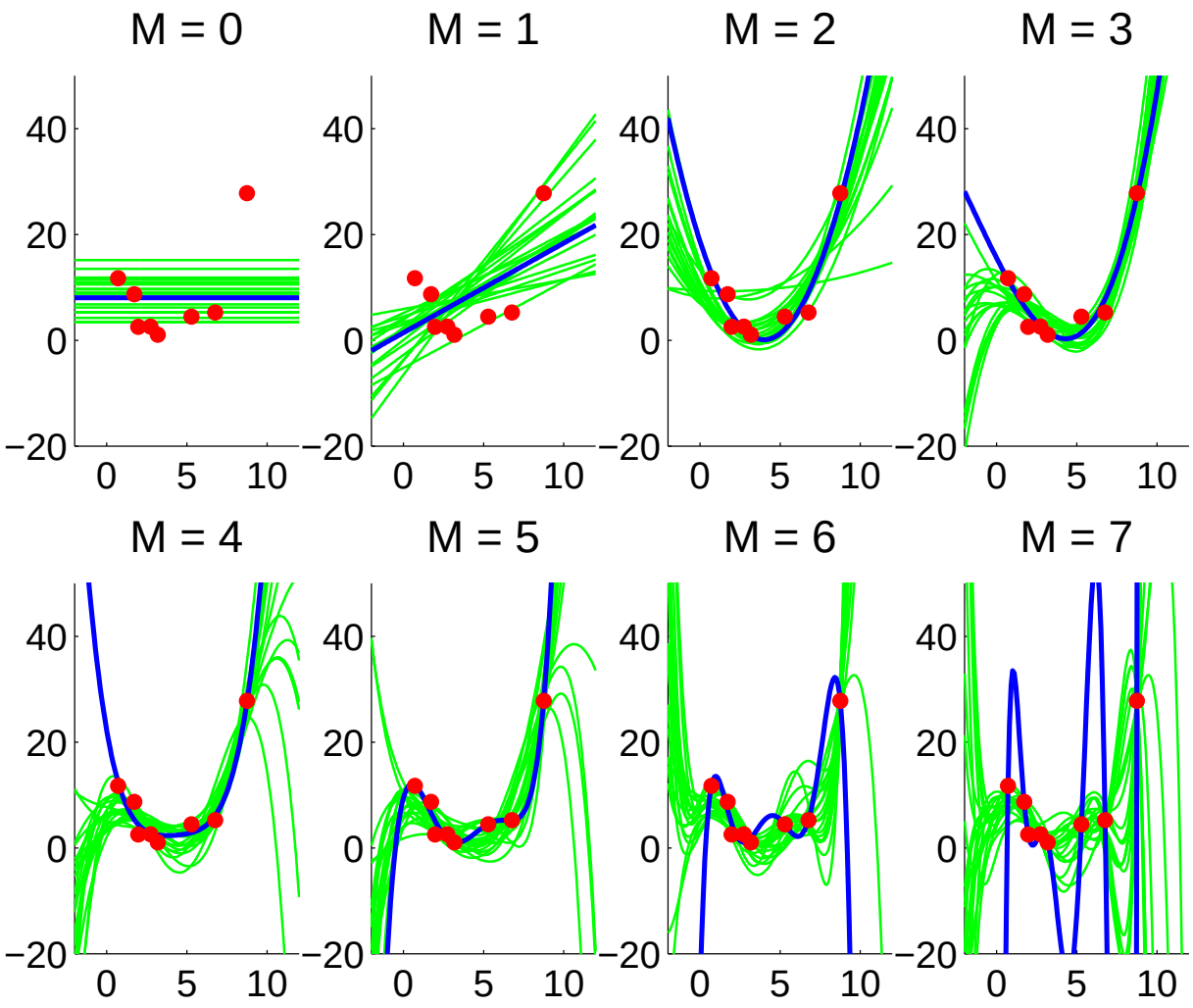
$$P(\mathcal{M} \,|\, \mathcal{D}) \;=\; \frac{P(\mathcal{D} \,|\, \mathcal{M})\; P(\mathcal{M})}{P(\mathcal{D})}$$

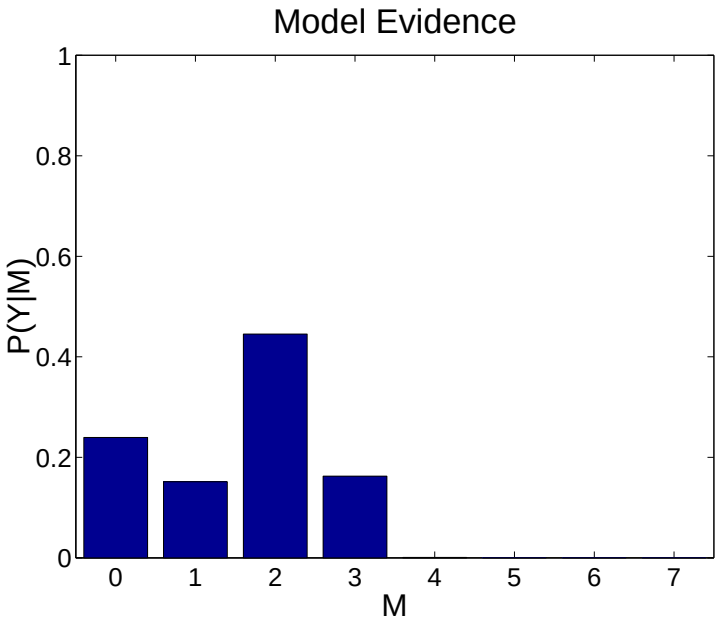
# Bayesian model comparison




Bishop "Pattern recognition and machine learning" (based on MacKay, 1995). See also Murray and Ghahramani (2005).

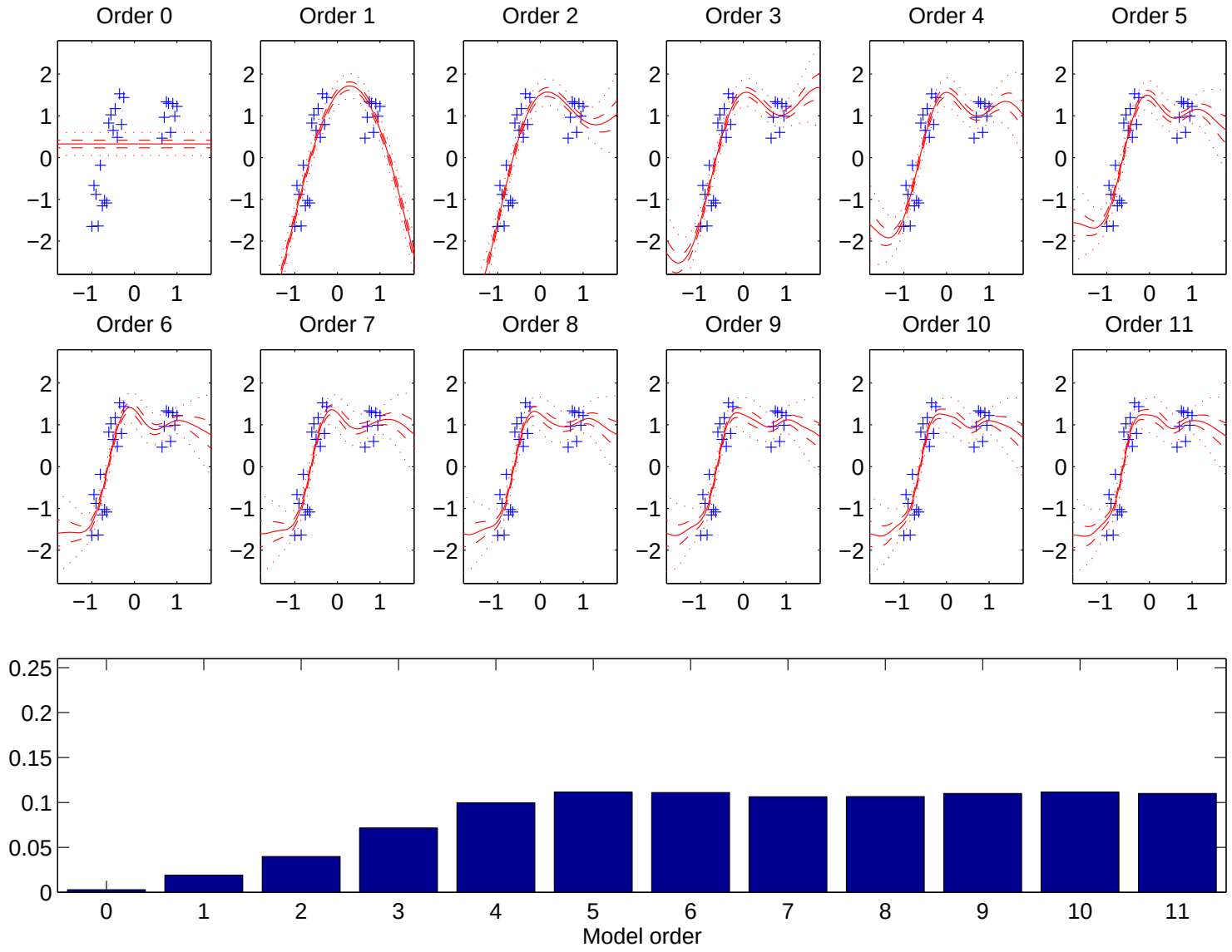
# Bayesian model comparison



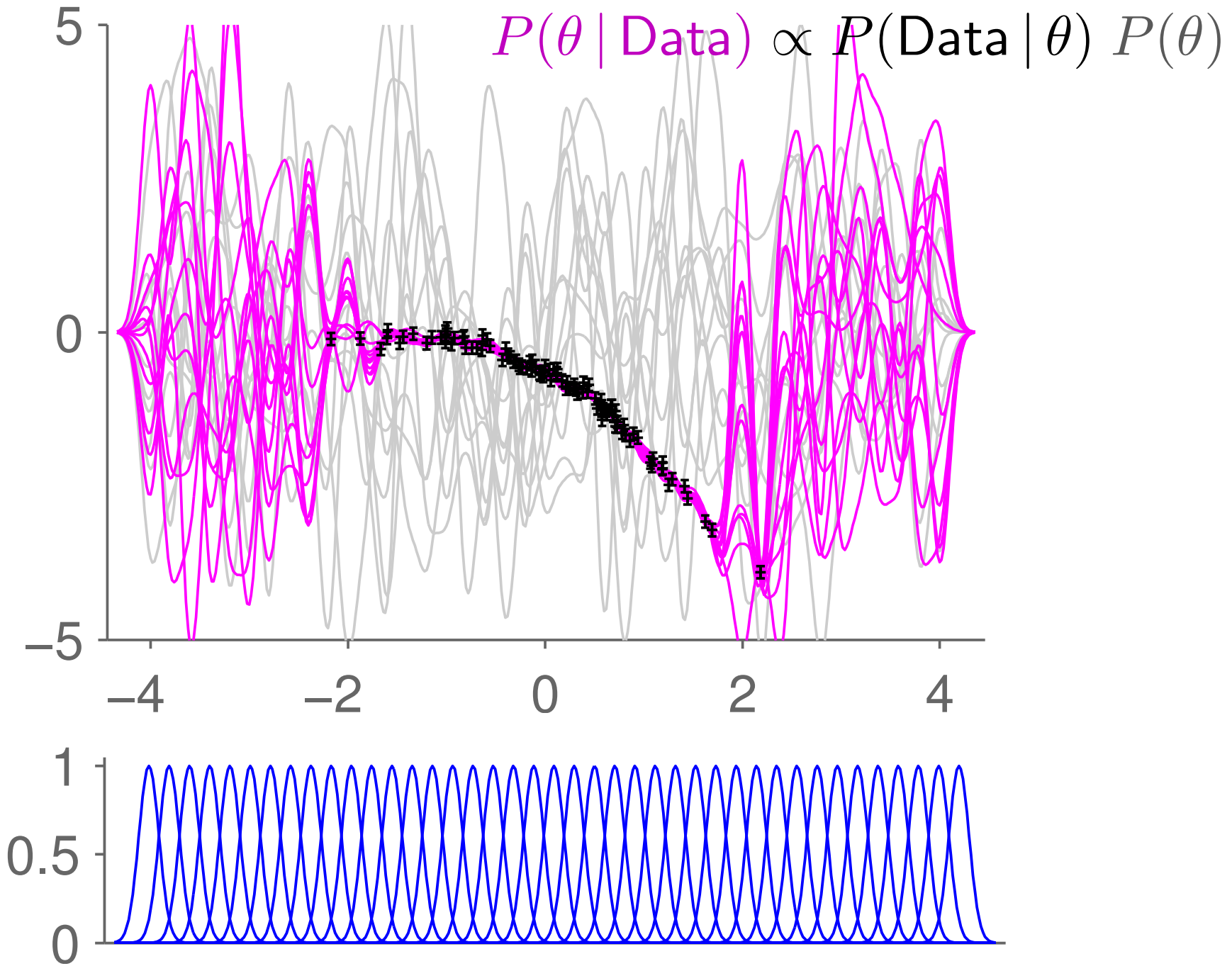

Demo stolen from Zoubin Ghahramani's notes

# Bayesian model comparison




Rasmussen and Ghahramani (NIPS, 2001)

# Bayesian linear regression (with RBF features)



# Summary

Some Machine Learning needs inference with uncertainty

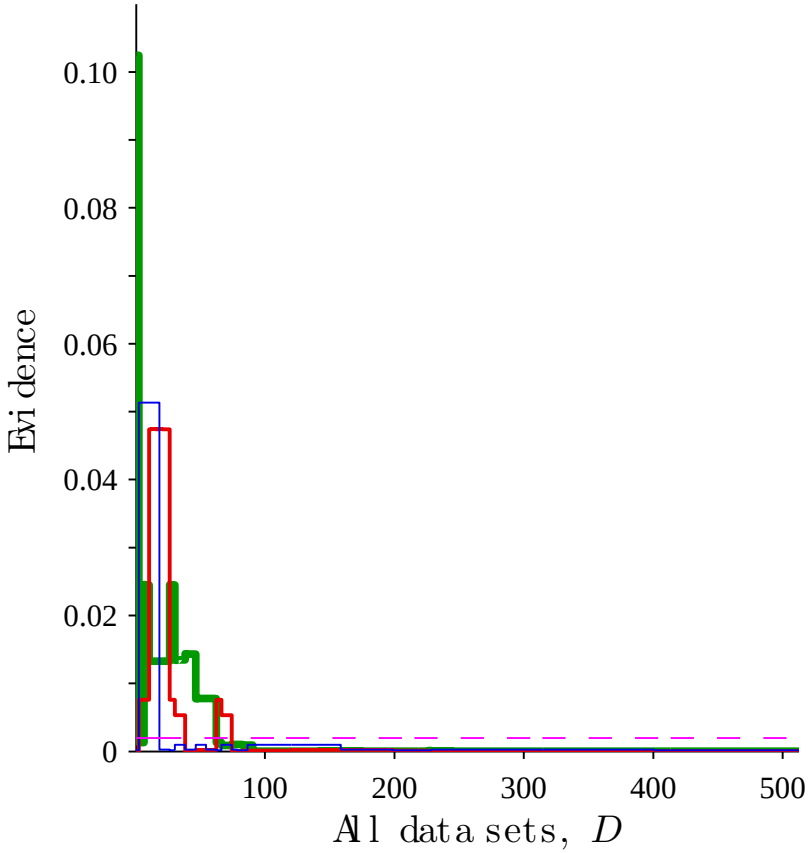
Model selection isn't to avoid overfitting:

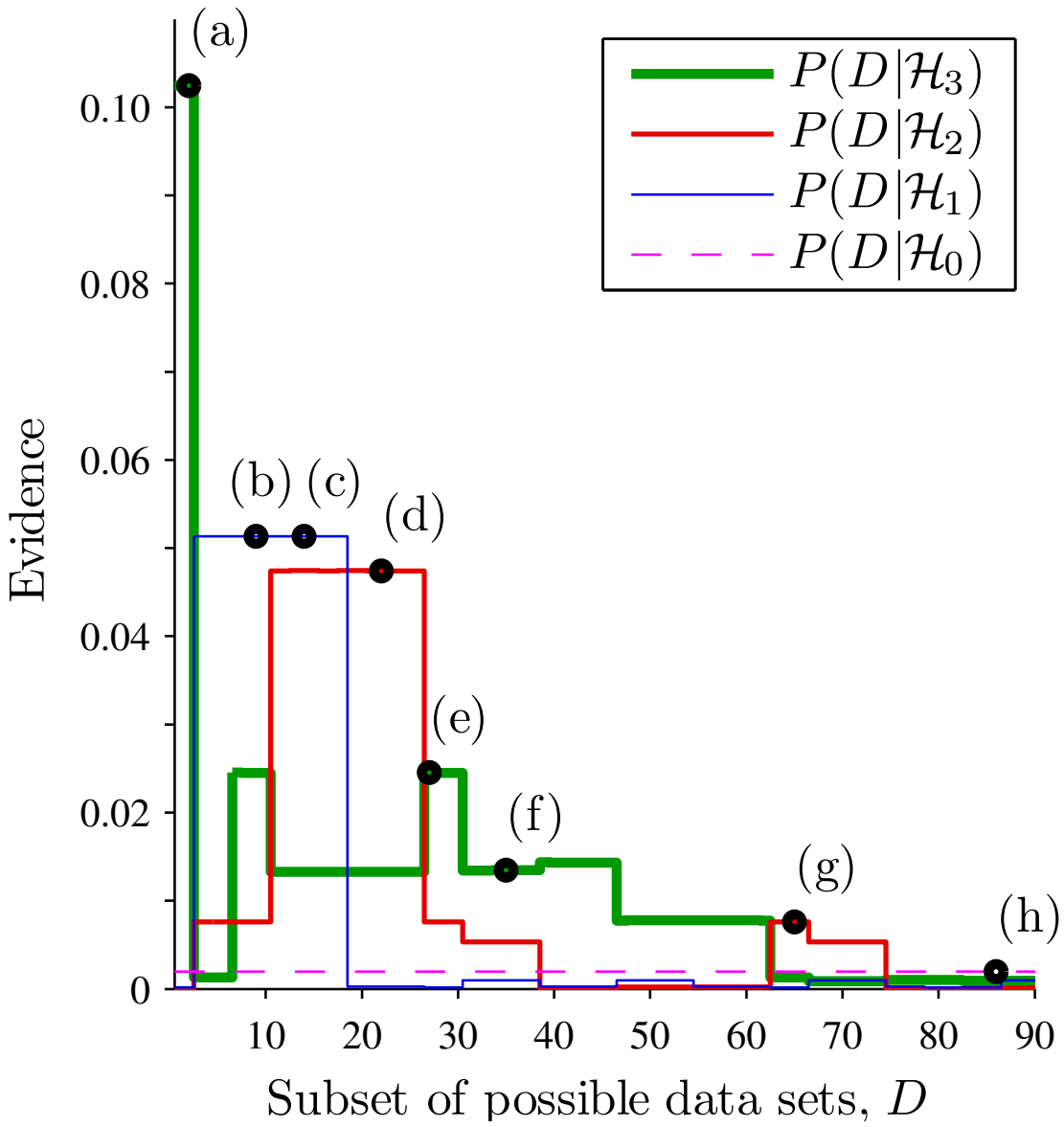
- can need many parameters with little data

Nonparametric models must contain strong structure!

- complexity often controlled with hyperparameters

# The messy truth



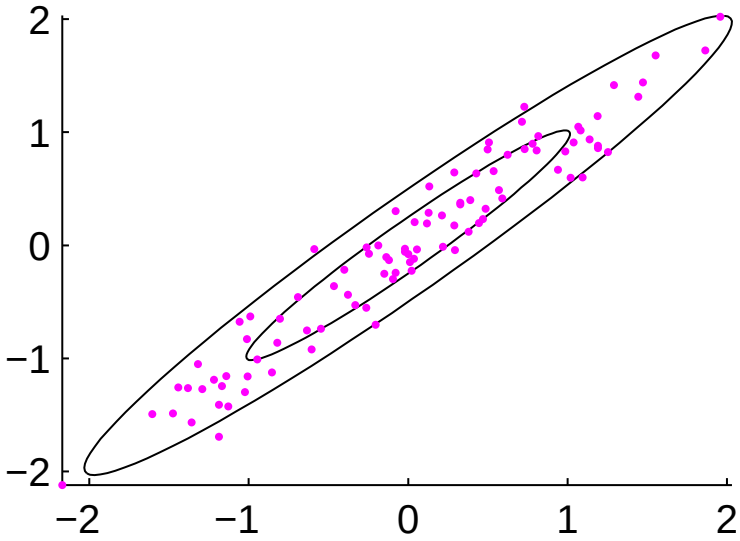




Murray and Ghahramani (2005).

# Gaussian Processes

Many inference problems involve high-dimensional integrals

**We're really good at integrating Gaussians(!)**




Can we really solve significant machine learning problems with a multivariate Gaussian?

# Gaussian distributions

Completely described by mean $\mu$ and covariance $\Sigma$:

$$P(\mathbf{f}|\Sigma,\mu) = |2\pi\Sigma|^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2}(\mathbf{f}-\mu)^T\Sigma^{-1}(\mathbf{f}-\mu)\right)$$

Where $\Sigma_{ij} = \langle f_i f_j \rangle - \mu_i \mu_j$

If we know a distribution is Gaussian and know its mean and covariances, we know its density function.

# Marginal of Gaussian

The marginal of a Gaussian distribution is Gaussian.

$$P(\mathbf{f}, \mathbf{g}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \begin{bmatrix} A & C \\ C^\top & B \end{bmatrix}\right)$$

As soon as you convince yourself that the marginal

$$P(\mathbf{f}) = \int \mathrm{d}\mathbf{g}\, P(\mathbf{f}, \mathbf{g})$$

is Gaussian, you already know the means and covariances:

$$P(\mathbf{f}) = \mathcal{N}(\mathbf{a}, A).$$

# Conditional of Gaussian

Any conditional of a Gaussian distribution is also Gaussian:

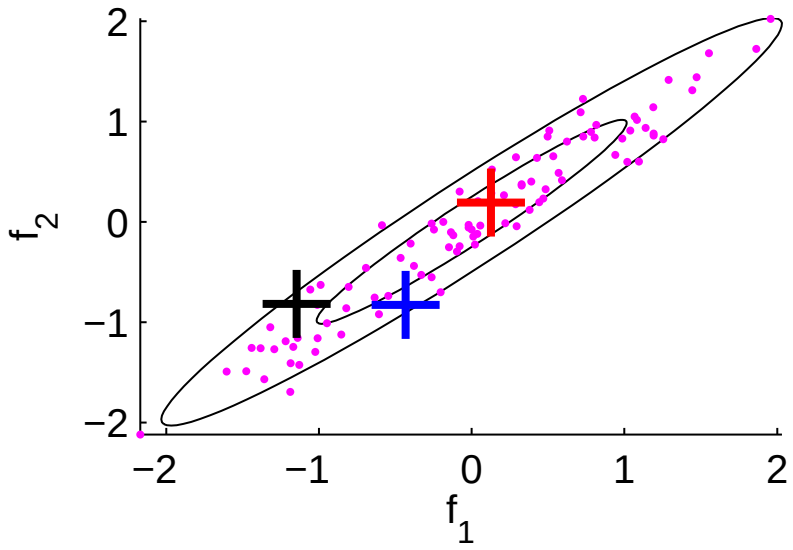
$$P(\mathbf{f}, \mathbf{g}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \begin{bmatrix} A & C \\ C^\top & B \end{bmatrix}\right)$$

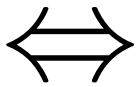
$$P(\mathbf{f}|\mathbf{g}) = \mathcal{N}(\mathbf{a} + CB^{-1}(\mathbf{y} - \mathbf{b}),\, A - CB^{-1}C^\top)$$

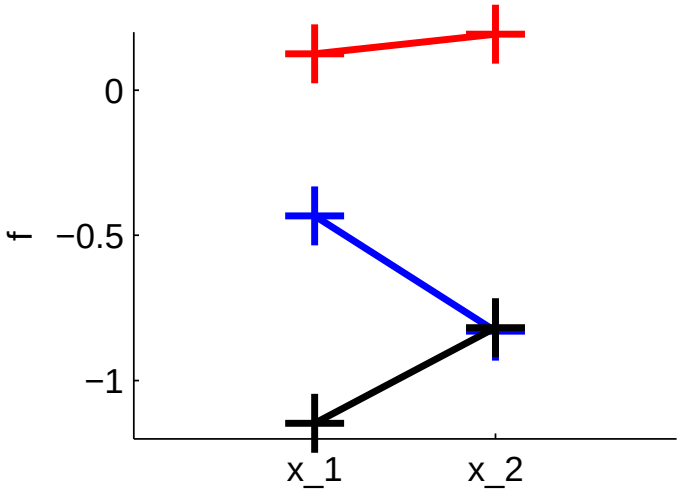
Showing this is not completely straightforward.
But it is a standard result, easily looked up.

# Laying out Gaussians

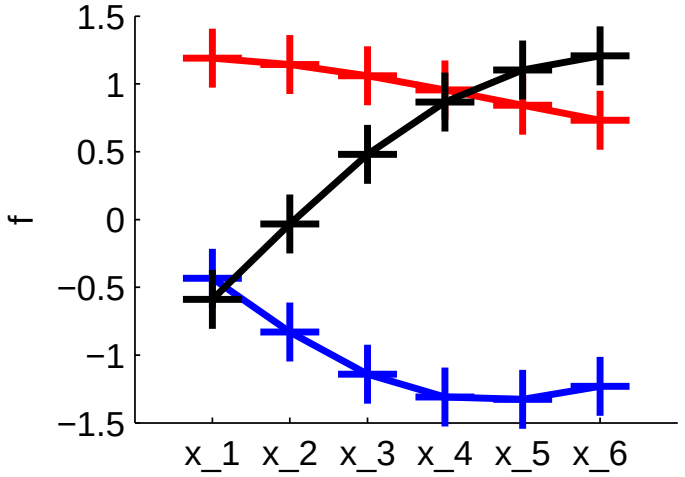
A way of visualizing draws from a 2D Gaussian:



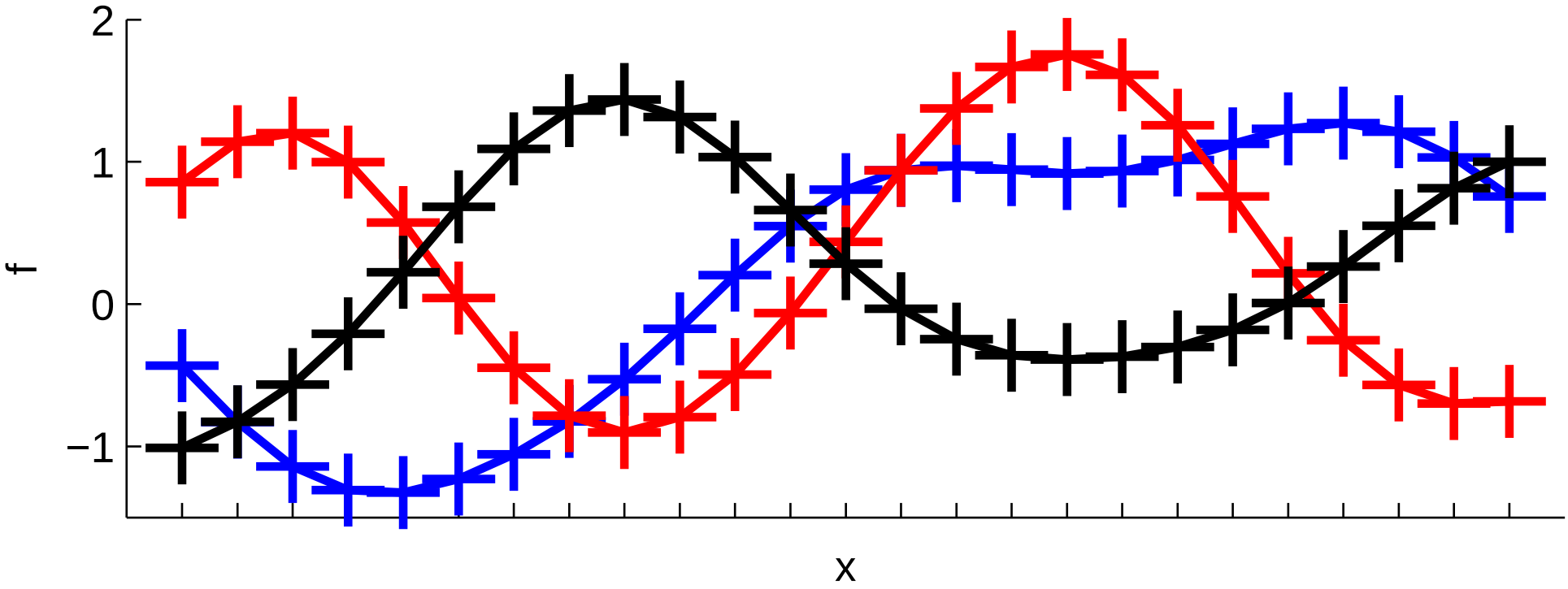



Now it's easy to show three draws from a 6D Gaussian:



# Building large Gaussians

Three draws from a 25D Gaussian:

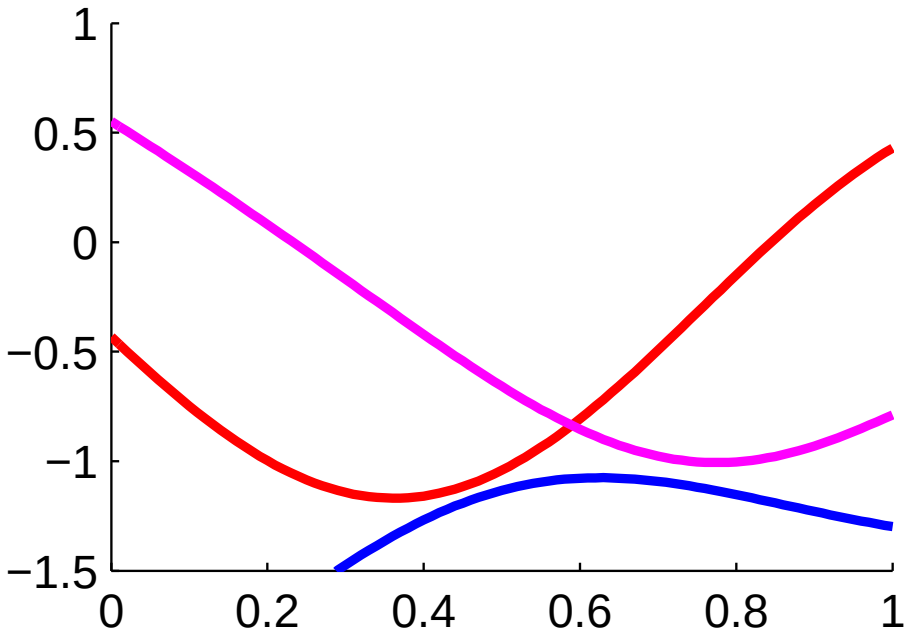



To produce this, we needed a mean: I used `zeros(25,1)`

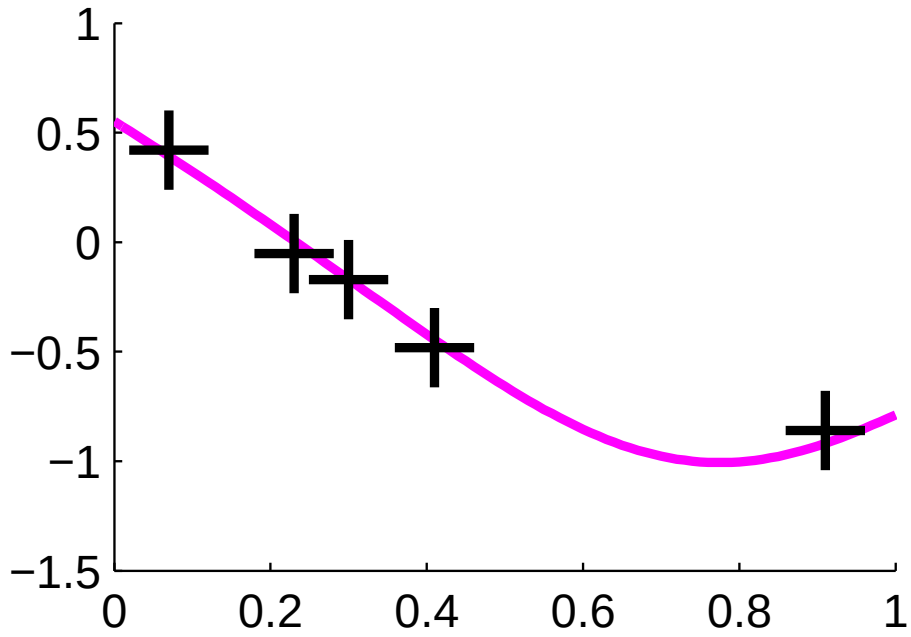
The covariances were set using a *kernel* function: $\Sigma_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

The $\mathbf{x}$'s are the positions that I planted the tics on the axis.

# GP regression model




$$f \sim \mathcal{GP}$$




Noisy observations:

$$y_i | f_i \sim \mathcal{N}(f_i, \sigma_n^2)$$

$$\mathbf{f} \sim \mathcal{N}(0, K), \quad K_{ij} = k(x_i, x_j)$$

where $f_i = f(x_i)$

# GP Posterior

Our prior over observations and targets is Gaussian:

$$P\left(\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix}\right) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) + \sigma_n^2 \mathbb{I} & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right)$$

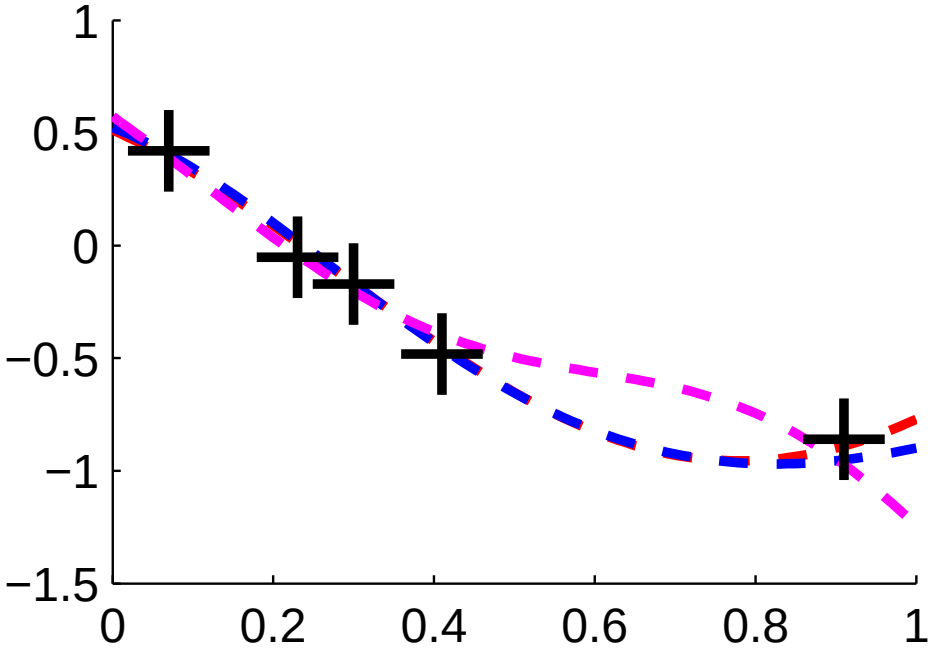
Using the rule for conditionals, $P(\mathbf{f}_*|\mathbf{y})$ is Gaussian with:

$$\text{mean}, \bar{\mathbf{f}}_* = K(X_*,X)(K(X,X) + \sigma_n^2\mathbb{I})^{-1}\mathbf{y}$$
$$\text{cov}(\mathbf{f}_*) = K(X_*,X_*) - K(X_*,X)(K(X,X) + \sigma_n^2\mathbb{I})^{-1}K(X,X_*)$$

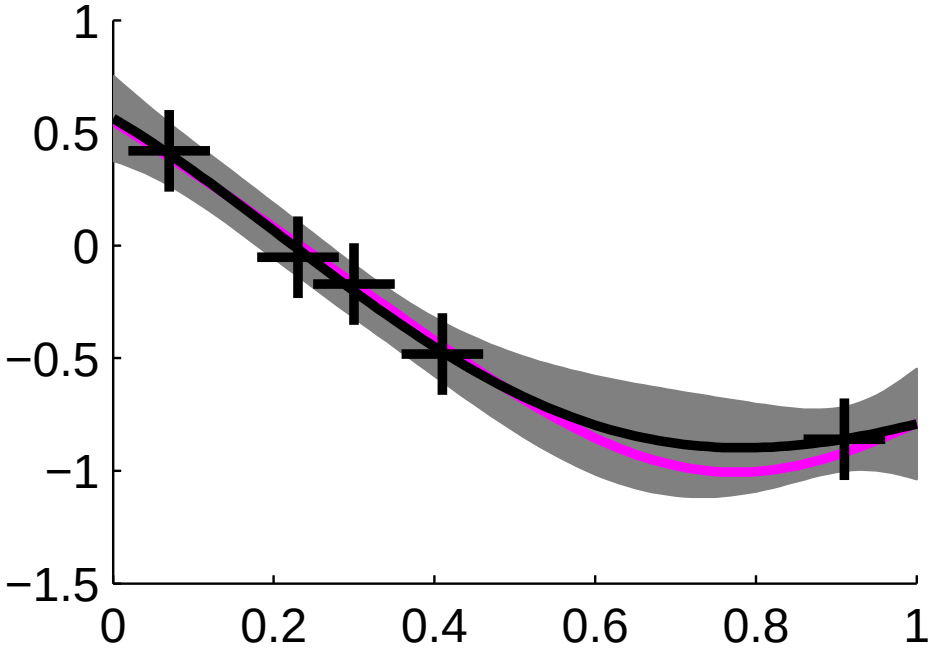
The posterior over functions is a Gaussian Process.

# GP posterior

Two (incomplete) ways of visualizing what we know:




Draws $\sim p(\mathbf{f}|\mathrm{data})$




Mean and error bars

# Gaussian Process Summary

We can represent a function as a *big* vector $\mathbf{f}$

We assume that this unknown vector was drawn from a *big* correlated Gaussian distribution, a *Gaussian process*.

(This might upset some mathematicians, but for all practical machine learning and statistical problems, this is fine.)

Observing elements of the vector (optionally corrupted by Gaussian noise) creates a posterior distribution. This is also Gaussian: the posterior over functions is still a Gaussian process.

Because marginalization in Gaussians is trivial, we can easily ignore all of the positions $\mathbf{x}_i$ that are neither observed nor queried.

# Appendix Slides

# Card prediction

**3 cards** with coloured faces:

1. one white and one black face
2. two black faces
3. two white faces

I shuffle the cards and flip them randomly. I select a card and way-up uniformly at random and place it on a table.

**Question:** You see a black face. What is the probability that the other side of the same card is white?

$$P(x_2 = \mathtt{W} \,|\, x_1 = \mathtt{B}) = \quad {}^1\!/_3, \quad {}^1\!/_2, \quad {}^2\!/_3, \quad \text{other?}$$

**Notes on the card prediction problem:**

This card problem is Ex. 8.10a), MacKay, p142.
`http://www.inference.phy.cam.ac.uk/mackay/itila/`

It is *not* the same as the famous 'Monty Hall' puzzle: Ex. 3.8–9 and
`http://en.wikipedia.org/wiki/Monty_Hall_problem`

The Monty Hall problem is also worth understanding. Although the card problem is (hopefully) less controversial and more straightforward. The process by which a card is selected should be clear: $P(c) = 1/3$ for $c = 1, 2, 3$, and the face you see first is chosen at random: e.g., $P(x_1 = \mathtt{B} | c = 1) = 0.5$.

Many people get this puzzle wrong on first viewing (it's easy to mess up). If you do get the answer right immediately (are you sure?), this is will be a simple example on which to demonstrate some formalism.

# How do we solve it formally?

**Use Bayes rule?**

$$P(x_2 = \mathtt{W} \mid x_1 = \mathtt{B}) = \frac{\boxed{P(x_1 = \mathtt{B} \mid x_2 = \mathtt{W})}\, P(x_2 = \mathtt{W})}{P(x_1 = \mathtt{B})}$$

The boxed term is no more obvious than the answer!

*Bayes rule is used to 'invert' forward generative processes that we understand.*

The first step to solve inference problems is to write down a model of your data.

# The card game model

**Cards:** 1) B|W, 2) B|B, 3) W|W

$$P(c) = \begin{cases} 1/3 & c = 1, 2, 3 \\ 0 & \text{otherwise.} \end{cases}$$

$$P(x_1 = \mathtt{B} \,|\, c) = \begin{cases} 1/2 & c = 1 \\ 1 & c = 2 \\ 0 & c = 3 \end{cases}$$

Bayes rule can 'invert' this to tell us $P(c \,|\, x_1 = \mathtt{B})$;
infer the generative process for the data we have.

# Inferring the card

**Cards:** 1) B|W, 2) B|B, 3) W|W

$$P(c \mid x_1 = \mathtt{B}) = \frac{P(x_1 = \mathtt{B} \mid c)\, P(c)}{P(x_1 = \mathtt{B})}$$

$$\propto \begin{cases} 1/2 \cdot 1/3 = 1/6 & c = 1 \\ 1 \cdot 1/3 = 1/3 & c = 2 \\ 0 & c = 3 \end{cases}$$

$$= \begin{cases} 1/3 & c = 1 \\ 2/3 & c = 2 \end{cases}$$

**Q** *"But aren't there two options given a black face, so it's 50–50?"*

**A** There are two options, but the likelihood for one of them is $2\times$ bigger

# Predicting the next outcome

For this problem we can spot the answer, for more complex problems we want a formal means to proceed.

$P(x_2 \,|\, x_1 = \mathtt{B})$?

Need to introduce $c$ to use expressions we know:

$$\begin{aligned} P(x_2 \,|\, x_1 = \mathtt{B}) &= \sum_{c \in 1,2,3} P(x_2, c \,|\, x_1 = \mathtt{B}) \\ &= \sum_{c \in 1,2,3} P(x_2 \,|\, x_1 = \mathtt{B}, c)\, P(c \,|\, x_1 = \mathtt{B}) \end{aligned}$$

Predictions we would make if we knew the card, weighted by the posterior probability of that card.

$P(x_2 = \mathtt{W} \,|\, x_1 = \mathtt{B}) = 1/3$

**Strategy for solving inference and prediction problems:**

When interested in something $y$, we often find we can't immediately write down mathematical expressions for $P(y \,|\, \text{data})$.

So we introduce stuff, $z$, that helps us define the problem:

$$P(y \,|\, \text{data}) = \sum_z P(y, z \,|\, \text{data})$$

by using the sum rule. And then split it up:

$$P(y \,|\, \text{data}) = \sum_z P(y \,|\, z, \text{data})\, P(z \,|\, \text{data})$$

using the product rule. If knowing extra stuff $z$ we can predict $y$, we are set: weight all such predictions by the posterior probability of the stuff ($P(z \,|\, \text{data})$, found with Bayes rule).

*Sometimes* the extra stuff summarizes everything we need to know to make a prediction:

$$P(y \,|\, z, \text{data}) = P(y \,|\, z)$$

although not in the card game above.

## Not convinced?

Not everyone believes the answer to the card game question.

Sometimes probabilities are counter-intuitive. I'd encourage you to write simulations of these games if you are at all uncertain. Here is an Octave/Matlab simulator I wrote for the card game question:

```
cards = [1 1;
         0 0;
         1 0];
num_cards = size(cards, 1);

N = 0; % Number of times first face is black
kk = 0; % Out of those, how many times the other side is white

for trial = 1:1e6
    card = ceil(num_cards * rand());
    face = 1 + (rand < 0.5);
    other_face = (face==1) + 1;
    x1 = cards(card, face);
    x2 = cards(card, other_face);

    if x1 == 0
        N = N + 1;
        kk = kk + (x2 == 1);
    end
end

approx_probability = kk / N
```