

Lecture 6

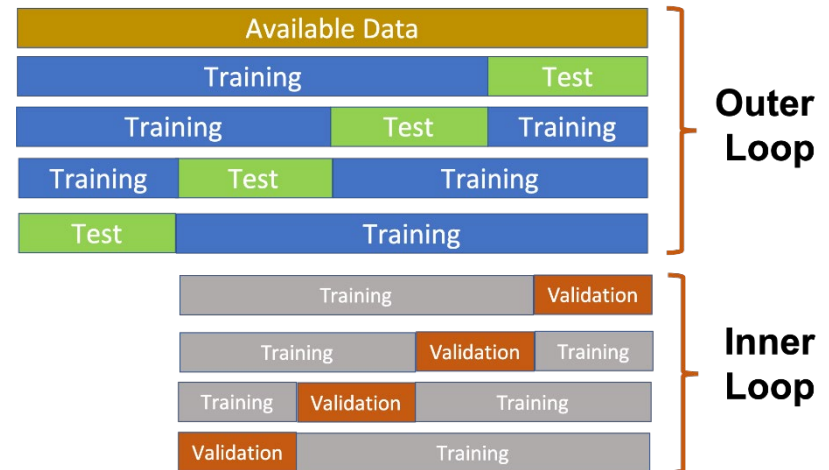
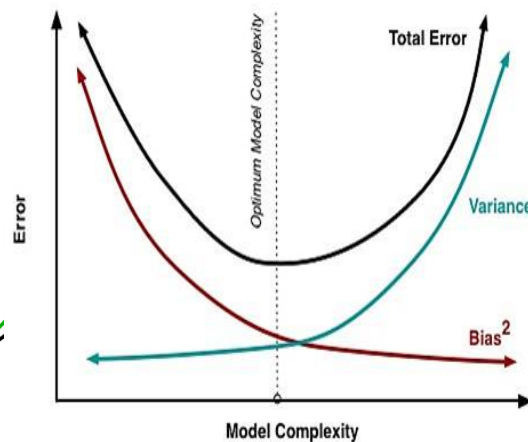
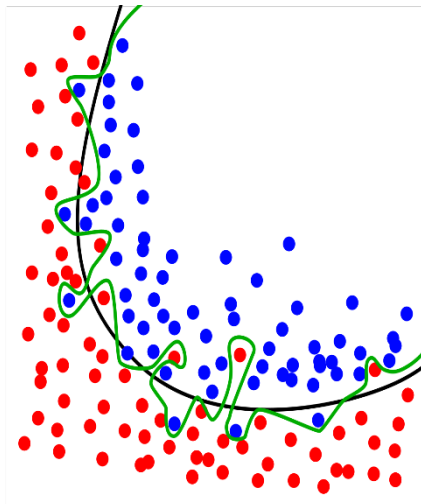
Bayesian Regression & Introduction to Deep Learning

Haiping Lu

COM4509/6509: Machine Learning & Adaptive Intelligence

Review of previous lecture(s)

- Linear regression: $y = mx + c$; $y = \mathbf{w}^T \mathbf{x} + b$; $\mathbf{y} = \mathbf{W} \mathbf{x} + \mathbf{b}$
- Overfitting, bias-variance trade-off, cross validation



Question

Our data is partitioned into three sets: training set, validation set, and test set. We train and tune our ML model on the validation set, getting an error E_{val} . We test this model on the test set, getting an error of E_{tst} .

*Which case is the mostly likely to be **overfitting**?*

A. $E_{\text{val}} \ll E_{\text{tst}}$

B. $E_{\text{val}} \gg E_{\text{tst}}$

C. $E_{\text{val}} = E_{\text{tst}}$

Week 6 Contents / Objectives

Part A

- Bayesian Inference
- Bayesian Linear Regression
- Predictive Distribution

Part B

- Computational Graph
- PyTorch: A Deep Learning Library

Week 6 Contents / Objectives

Part A

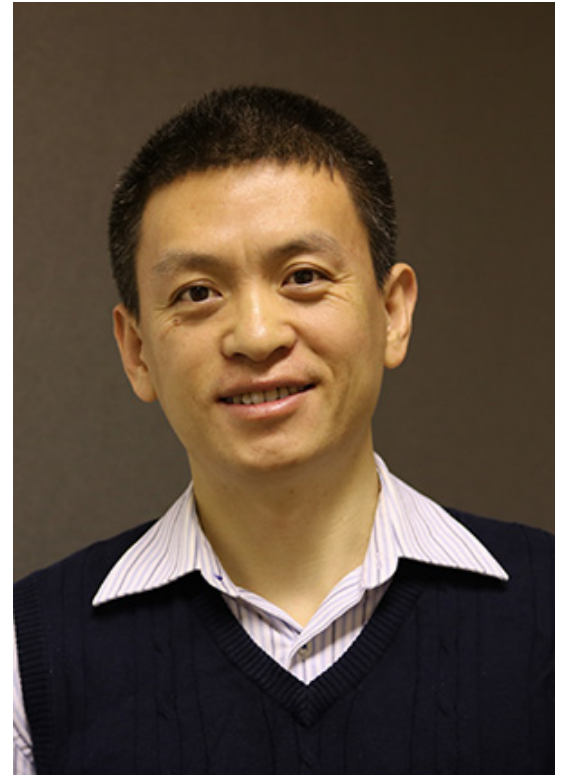
- **Bayesian Inference**
- Bayesian Linear Regression
- Predictive Distribution

Part B

- Computational Graph
- PyTorch: A Deep Learning Library

Question

- Which year was this photo taken?
 - A. 1996
 - B. 2006
 - C. 2016
 - D. 2026



Bayesian statistics

new data

$$p(y | \theta)$$

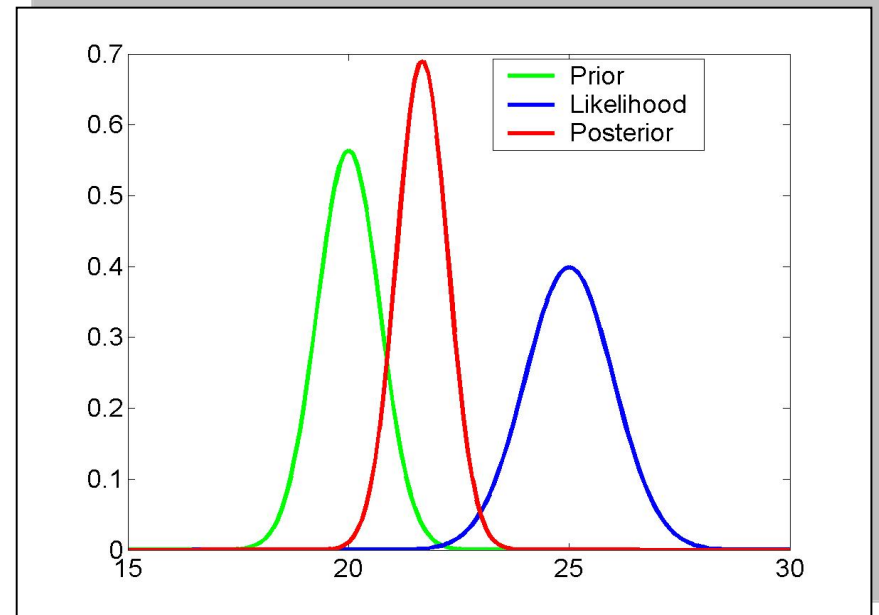
prior knowledge

$$p(\theta)$$

$$p(\theta | y) \propto p(y | \theta) p(\theta)$$

posterior \propto likelihood \cdot prior

Incorporate **prior knowledge** into computing statistical probabilities.



Posterior: combination of prior knowledge and new data, weighted by their relative *precision*

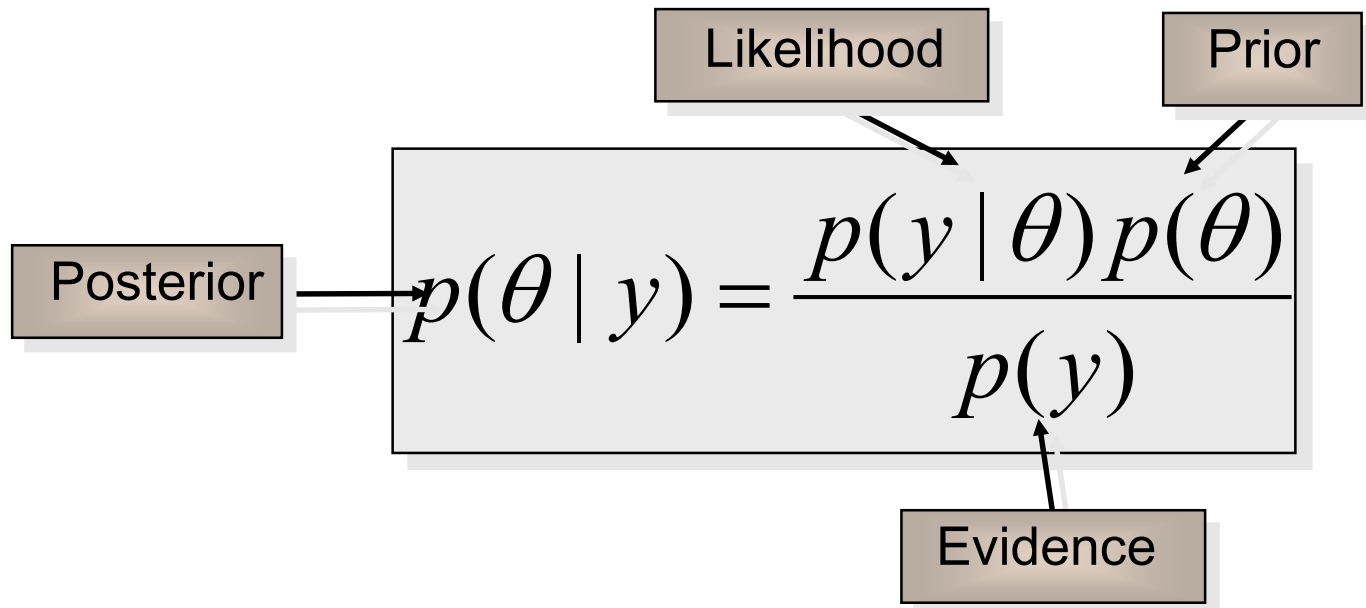
Bayes' rule

Given data y and parameters θ , their joint probability can be written as

$$p(\theta | y)p(y) = p(y, \theta)$$

$$p(y, \theta) = p(y | \theta)p(\theta)$$

Eliminating $p(y, \theta)$ gives Bayes' rule:



Key concepts

- **Prior** probability: the estimate of the probability of the model before the data (evidence) is observed.
- **Posterior** probability: the probability of the model after observing the data (evidence).
- **Likelihood**: the probability of observing a (random) data point given a model (*fixed*) → the compatibility of the data (evidence) with the given model.
- **Marginal likelihood**: "model **evidence**", the same for all possible model variations being considered

Principles of Bayesian inference

⇒ Formulation of a generative model

likelihood $p(y|\theta)$
prior distribution $p(\theta)$

⇒ Observation of data

y

⇒ Update of beliefs based upon observations, given a prior state of knowledge

$$p(\theta | y) \propto p(y | \theta)p(\theta)$$

Week 6 Contents / Objectives

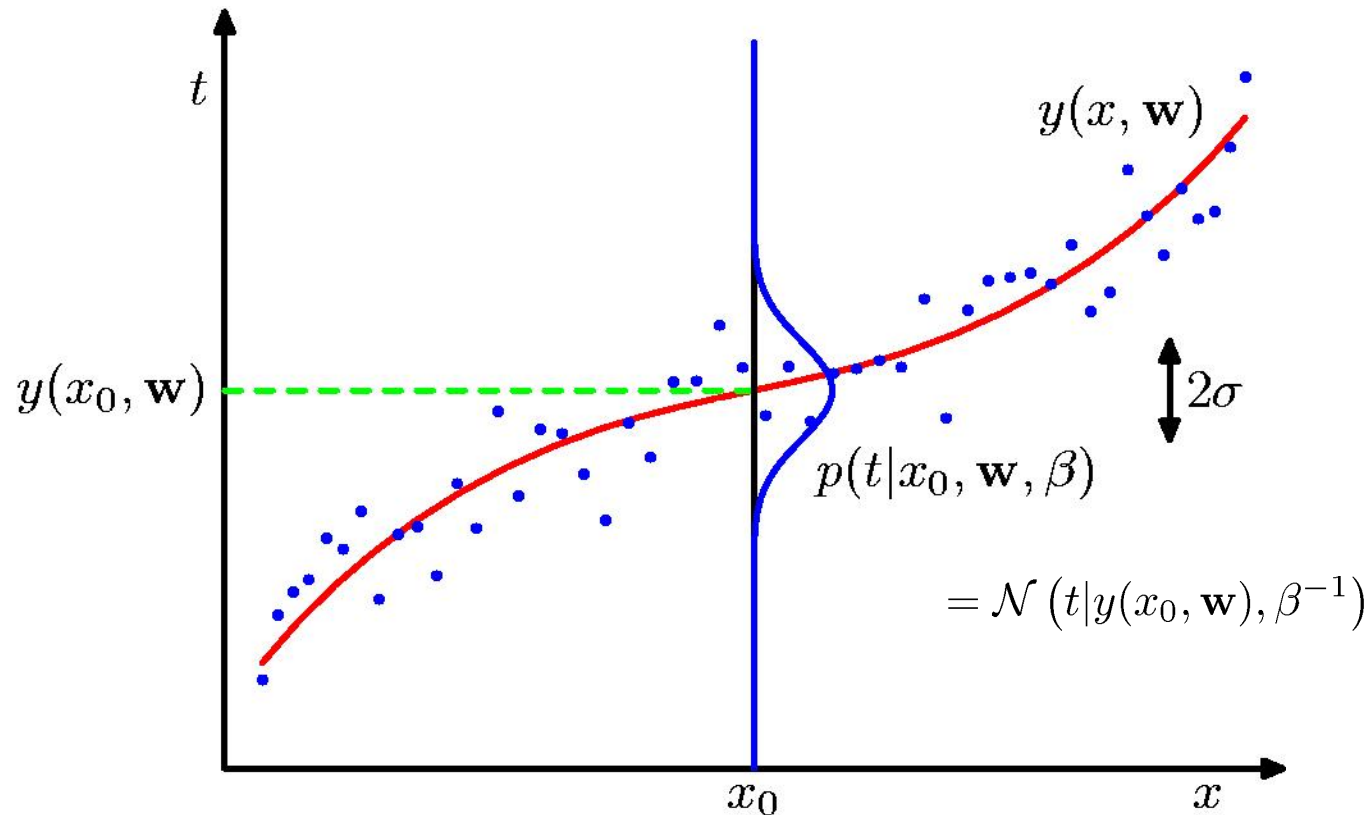
Part A

- Bayesian Inference
- **Bayesian Linear Regression**
- Predictive Distribution

Part B

- Computational Graph
- PyTorch: A Deep Learning Library

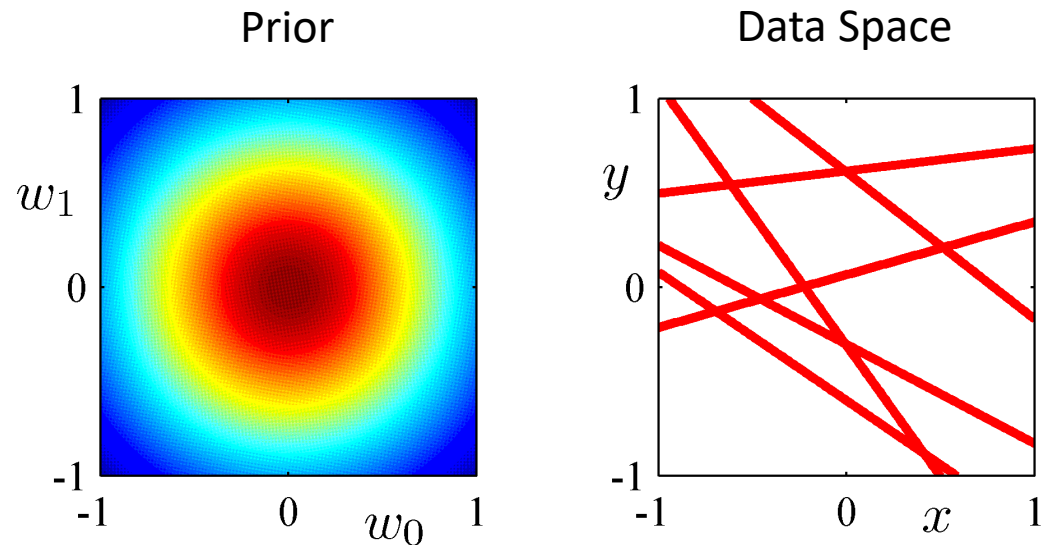
Linear Regression: Curve Fitting



- **Precision** $\beta = \frac{1}{\sigma^2}$; note: *notation difference*

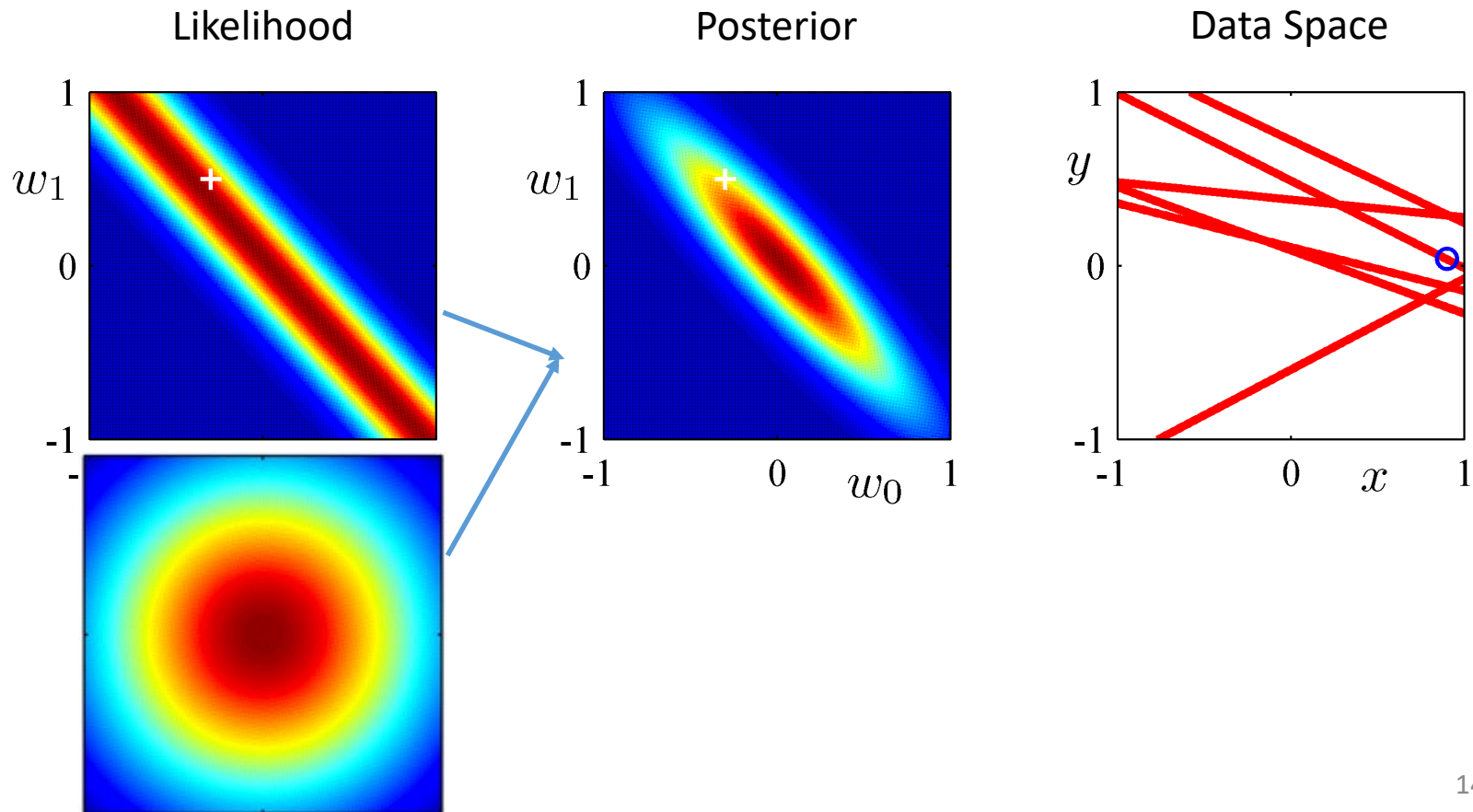
Bayesian Linear Regression (1)

0 data points observed. Six samples of $y(x, \mathbf{w})$ shown.



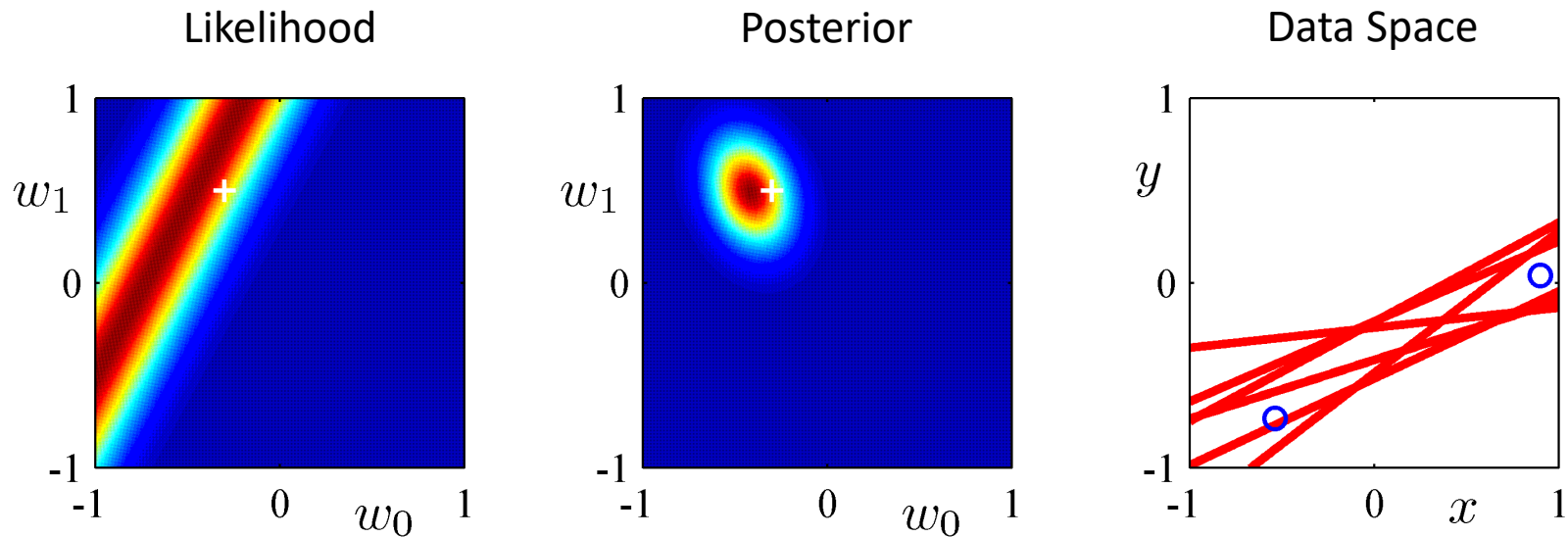
Bayesian Linear Regression (2)

1 data point observed \rightarrow soft constraint. This posterior \rightarrow prior for the next data point observed)



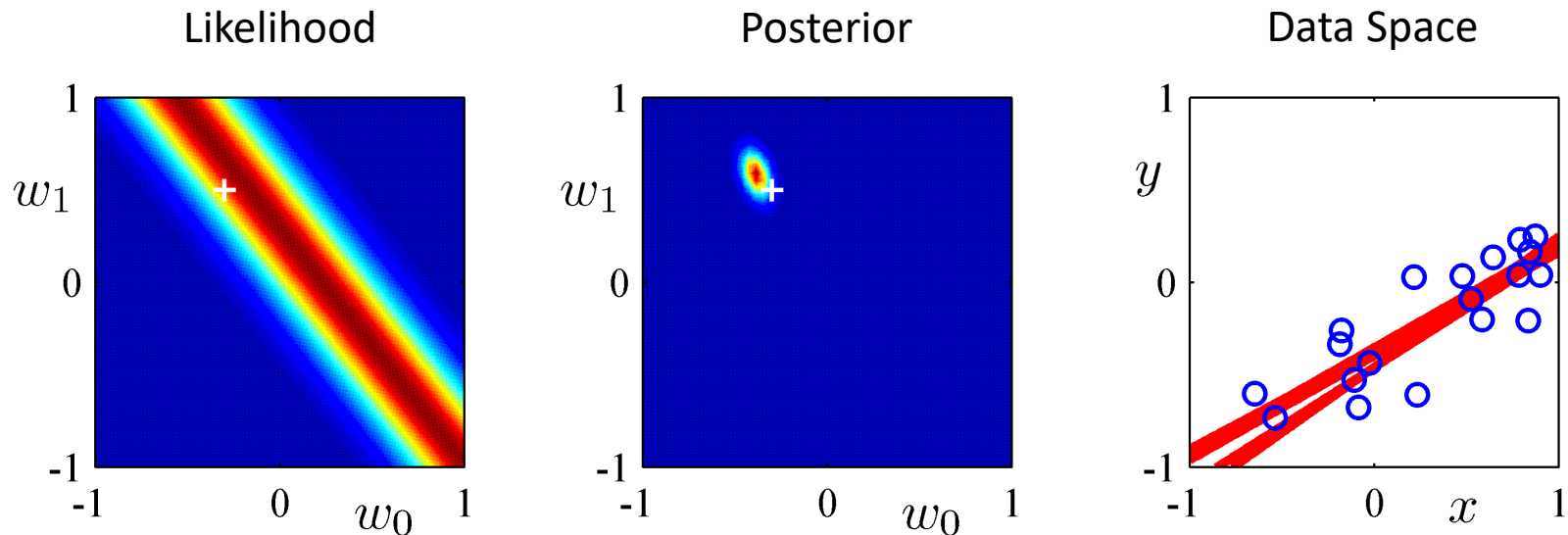
Bayesian Linear Regression (3)

2 data points observed



Bayesian Linear Regression (4)

20 data points observed



Univariate Gaussian: density est.

Normal densities

$$p(\beta) = N(\beta; \mu_p, \alpha_p^{-1})$$

$$p(y | \beta) = N(y; \beta, \alpha_e^{-1})$$

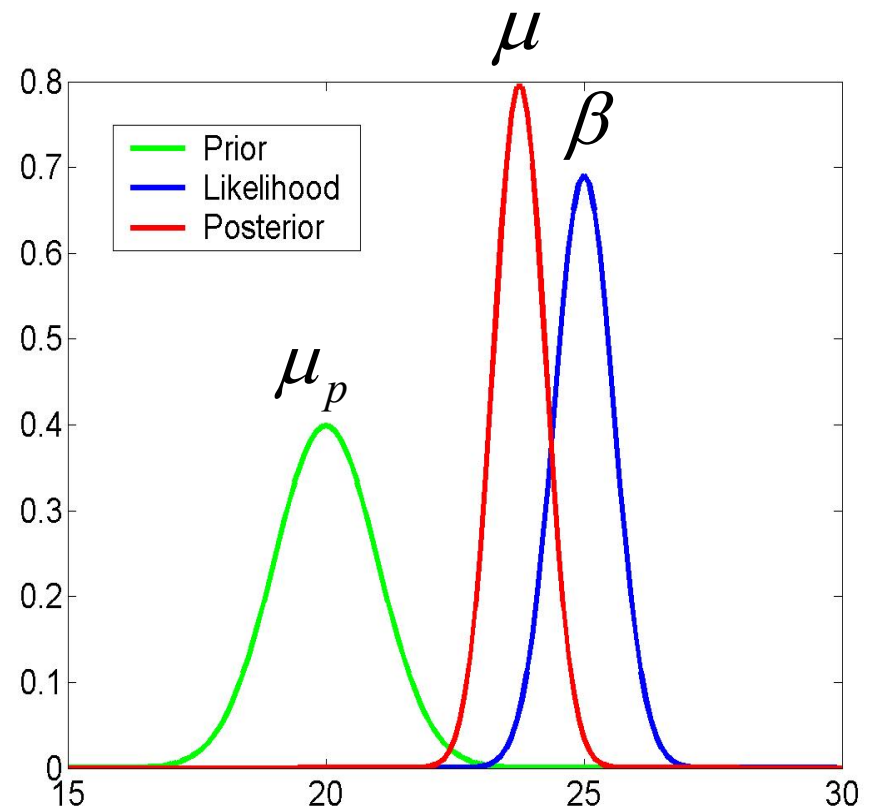
$$p(\beta | y) = N(\beta; \mu, \alpha^{-1})$$

$$\alpha = \alpha_e + \alpha_p$$

$$\mu = \alpha^{-1}(\alpha_e y + \alpha_p \mu_p)$$

Posterior mean =
precision-weighted combination of
prior mean and data mean

$$y = \beta + e$$



Bayesian regression: univariate

Normal densities

$$p(\beta) = N(\beta; \mu_p, \alpha_p^{-1})$$

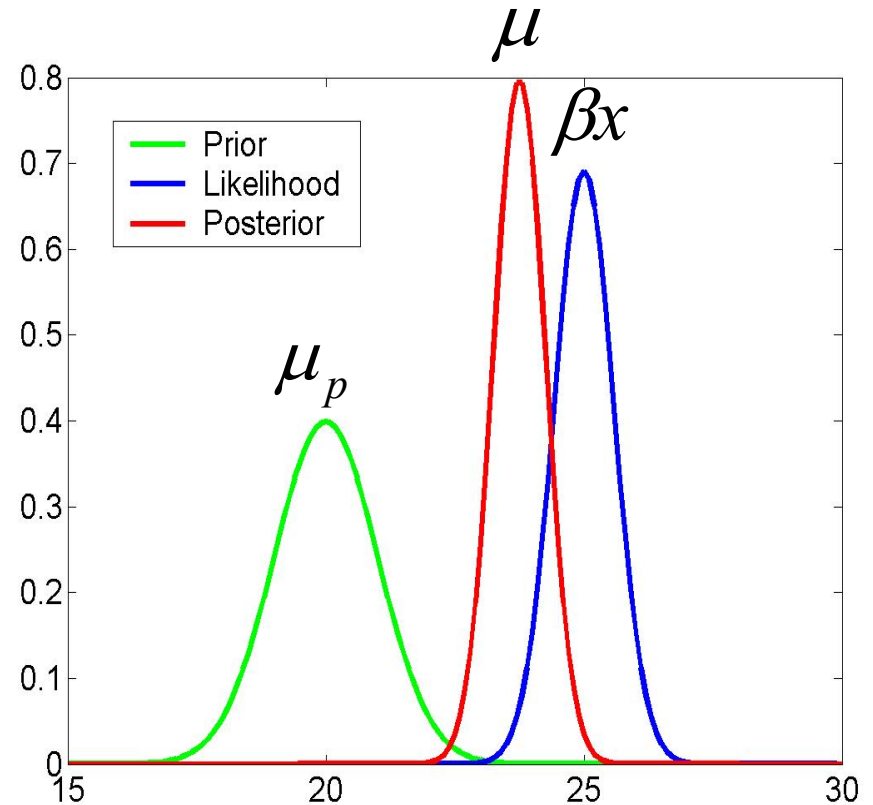
$$p(y | \beta) = N(y; \beta x, \alpha_e^{-1})$$

$$p(\beta | y) = N(\beta; \mu, \alpha^{-1})$$

$$\alpha = \alpha_e x^2 + \alpha_p$$

$$\mu = \alpha^{-1} (\alpha_e x y + \alpha_p \mu_p)$$

$$y = \beta x + e$$



Bayesian regression: multivariate

Normal densities

$$p(\beta) = N(\beta; \mu_p, C_p)$$

$$p(y | \beta) = N(y; X\beta, C_e)$$

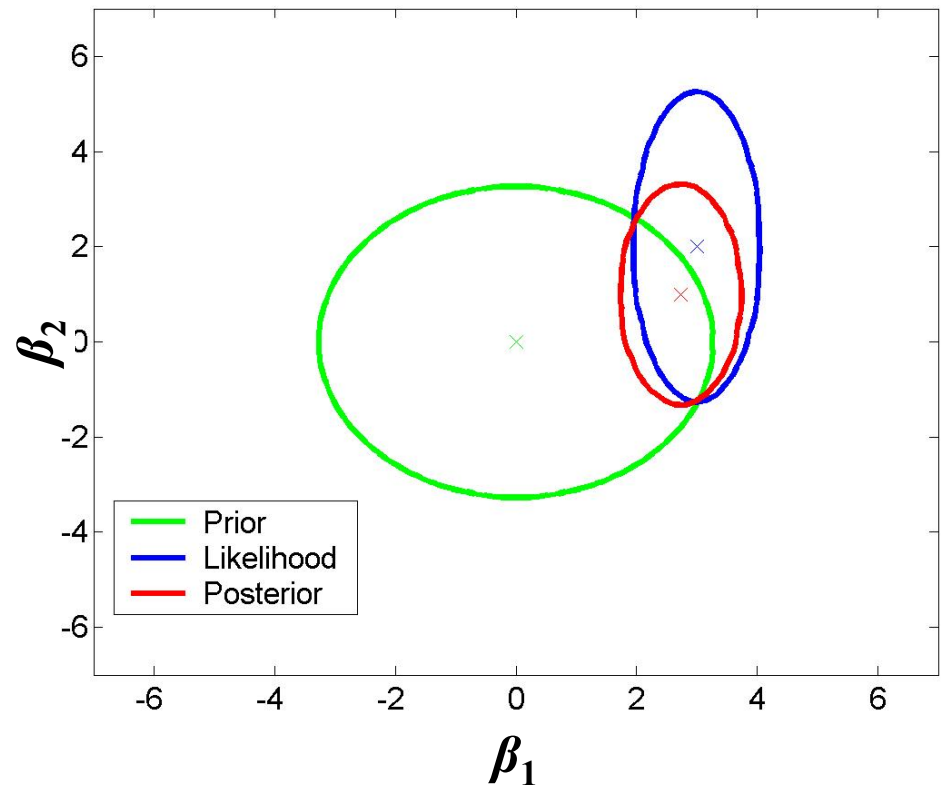
$$p(\beta | y) = N(\beta; \mu, C)$$

$$C^{-1} = X^T C_e^{-1} X + C_p^{-1}$$

$$\mu = C(X^T C_e^{-1} y + C_p^{-1} \mu_p)$$

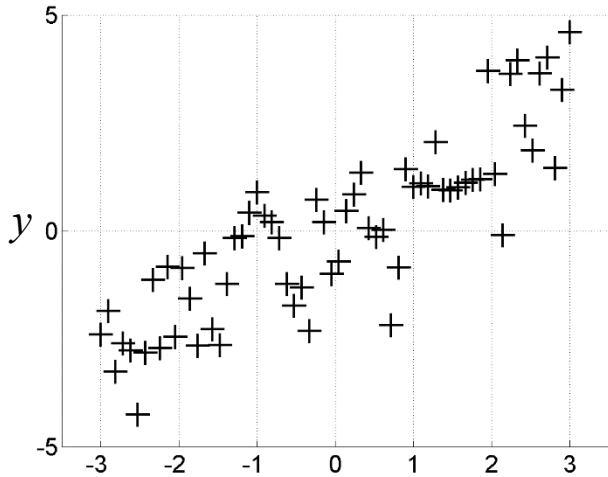
One step if C_e and C_p are known.
Otherwise iterative estimation
(EM).

$$y = X\beta + e$$



Linear regression

Data



Ordinary least squares

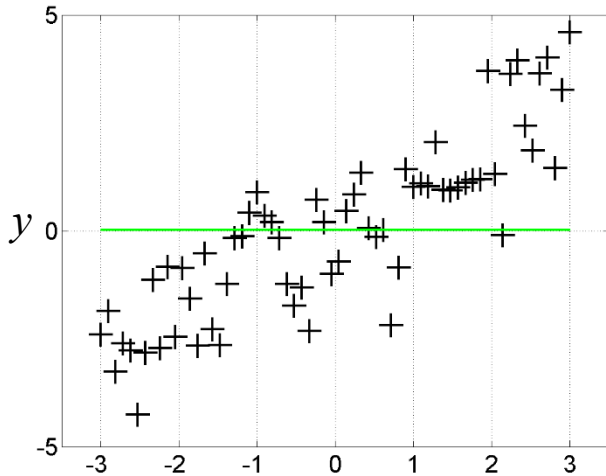
$$y = X\beta$$

$$E_D = (y - X\beta)^T (y - X\beta)$$

$$\frac{\partial E_D}{\partial \beta} = 0 \Rightarrow \hat{\beta}_{ols} = (X^T X)^{-1} X^T y$$

Linear regression

Data and model fit



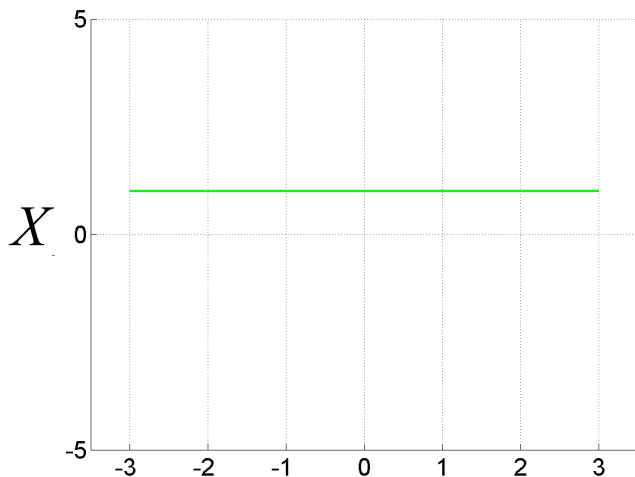
Ordinary least squares

$$y = X\beta$$

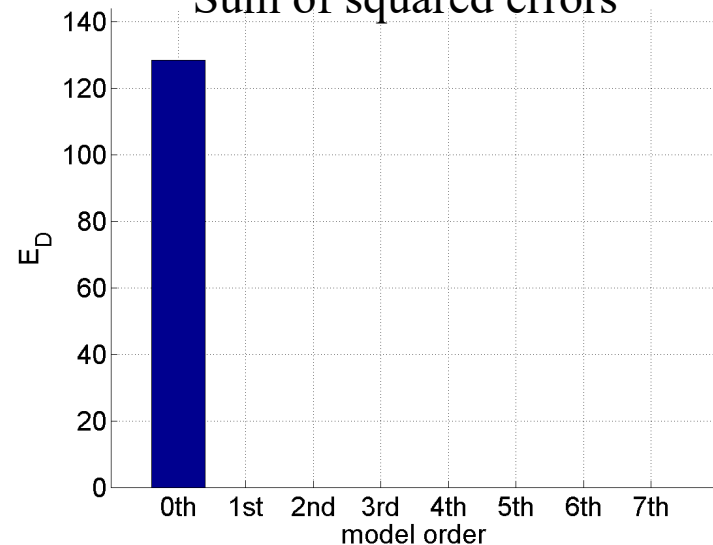
$$E_D = (y - X\beta)^T (y - X\beta)$$

$$\frac{\partial E_D}{\partial \beta} = 0 \Rightarrow \hat{\beta}_{ols} = (X^T X)^{-1} X^T y$$

Bases (explanatory variables)

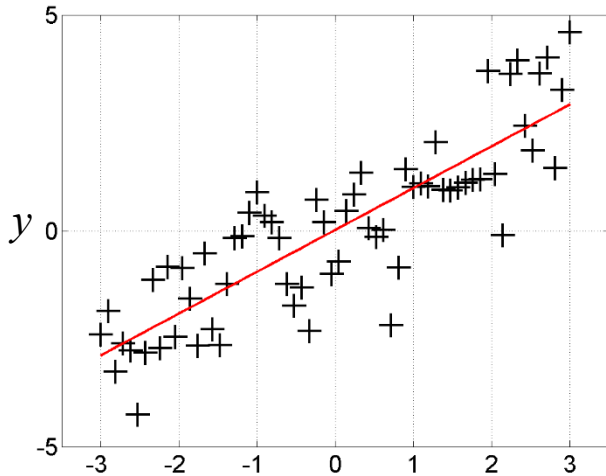


Sum of squared errors



Linear regression

Data and model fit



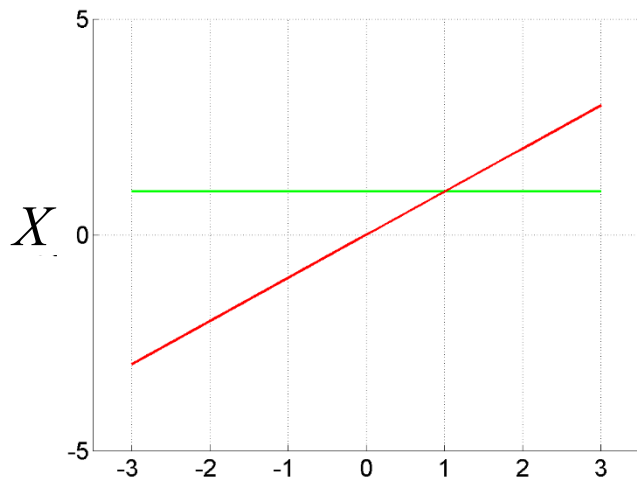
Ordinary least squares

$$y = X\beta$$

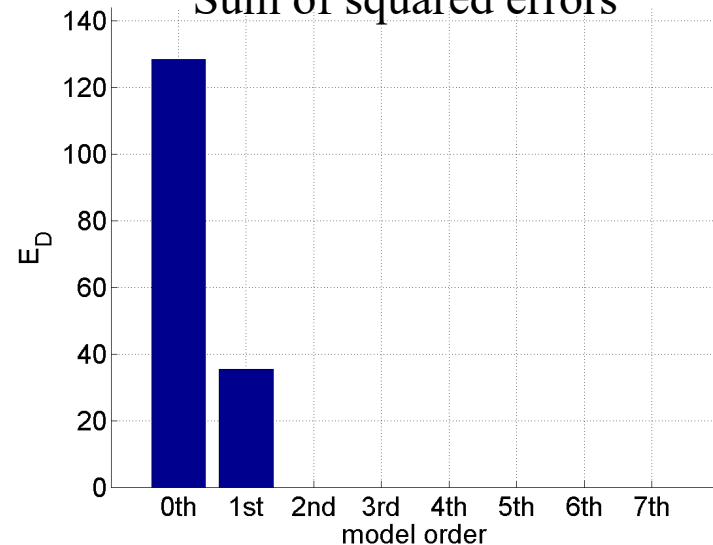
$$E_D = (y - X\beta)^T (y - X\beta)$$

$$\frac{\partial E_D}{\partial \beta} = 0 \Rightarrow \hat{\beta}_{ols} = (X^T X)^{-1} X^T y$$

Bases (explanatory variables)

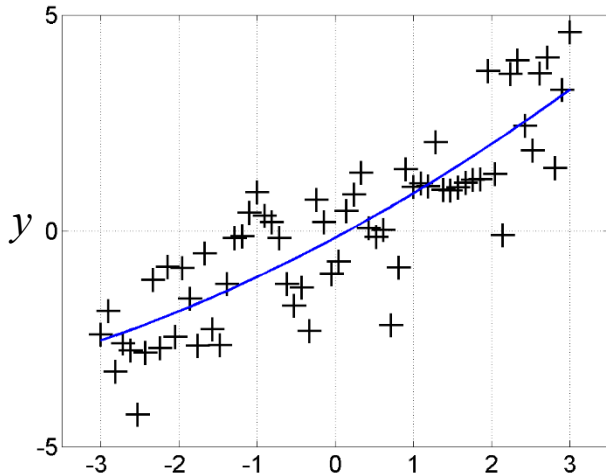


Sum of squared errors



Linear regression

Data and model fit



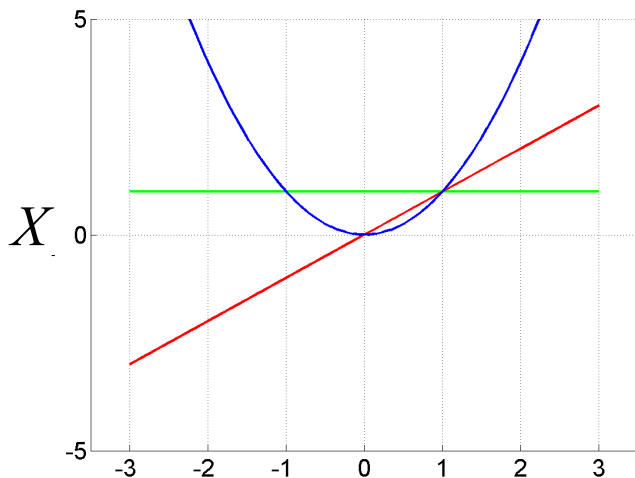
Ordinary least squares

$$y = X\beta$$

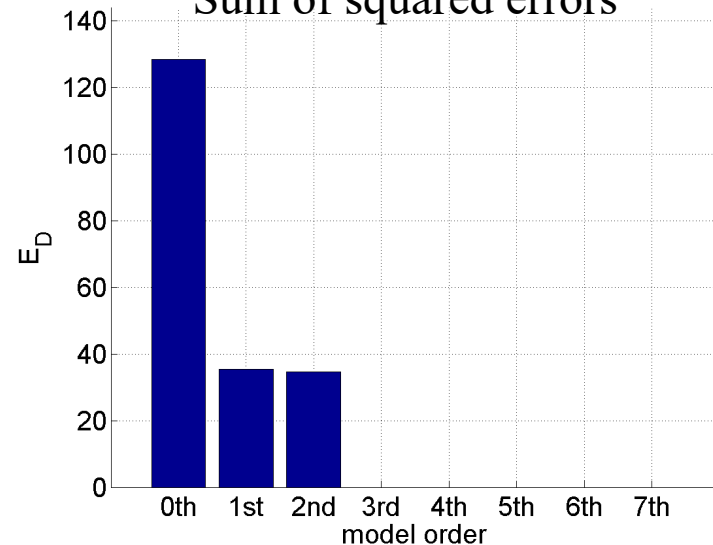
$$E_D = (y - X\beta)^T (y - X\beta)$$

$$\frac{\partial E_D}{\partial \beta} = 0 \Rightarrow \hat{\beta}_{ols} = (X^T X)^{-1} X^T y$$

Bases (explanatory variables)

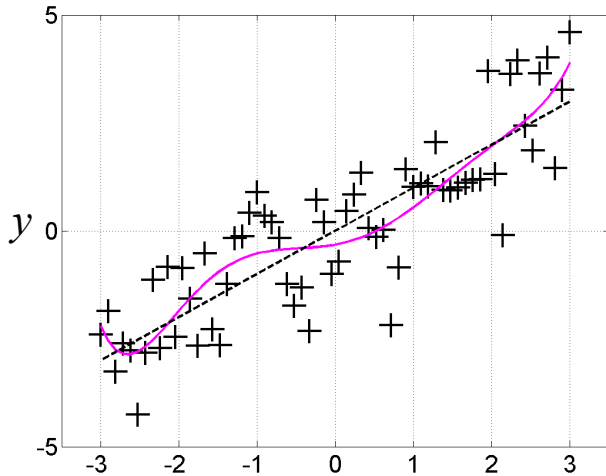


Sum of squared errors



Linear regression

Data and model fit

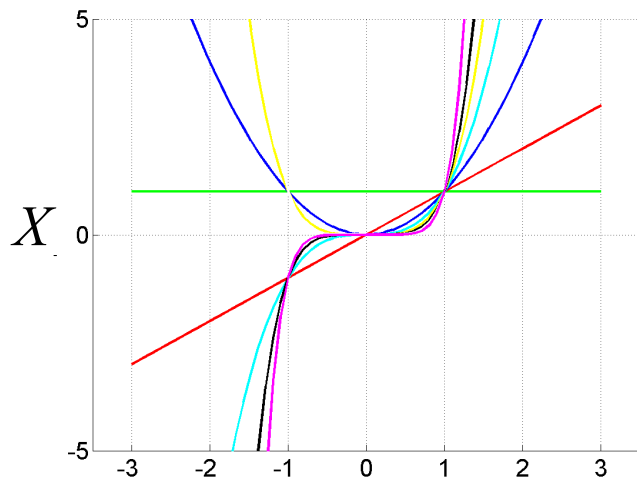


Ordinary least squares

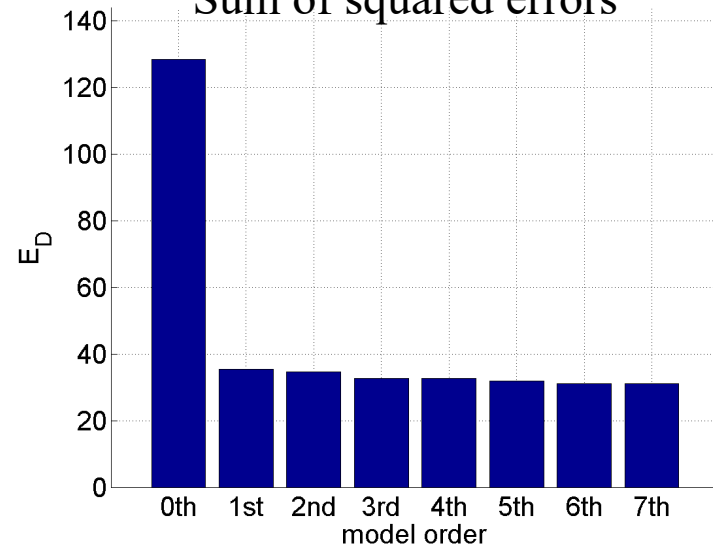
Over-fitting: model fits noise

Solution: include uncertainty in model parameters

Bases (explanatory variables)

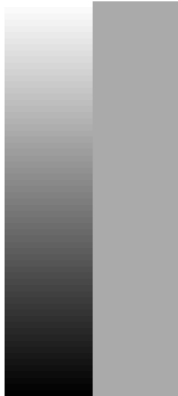


Sum of squared errors



Bayesian regression: *priors & likelihood*

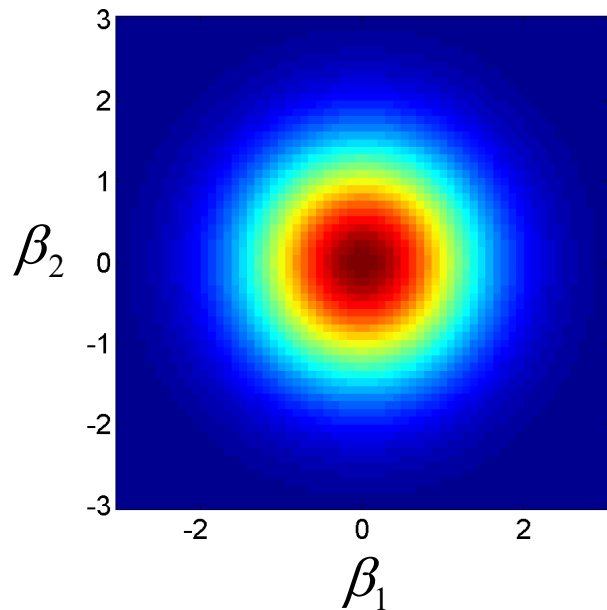
$X =$



Model:

$$y = X\beta + e$$

Bayesian regression: *priors & likelihood*



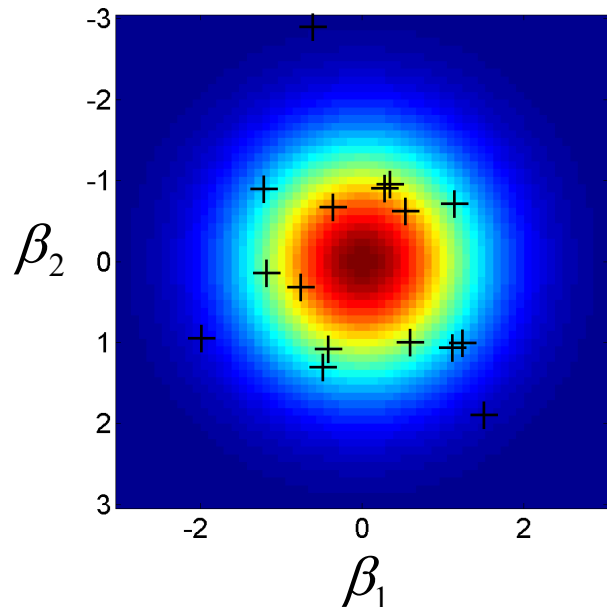
Model:

$$y = X\beta + e$$

Prior:

$$p(\beta|\alpha_2) = N_k(0, \alpha_2^{-1} I_k) \\ \propto \exp(-\alpha_2 \|\beta\|^2 / 2)$$

Bayesian regression: *priors & likelihood*

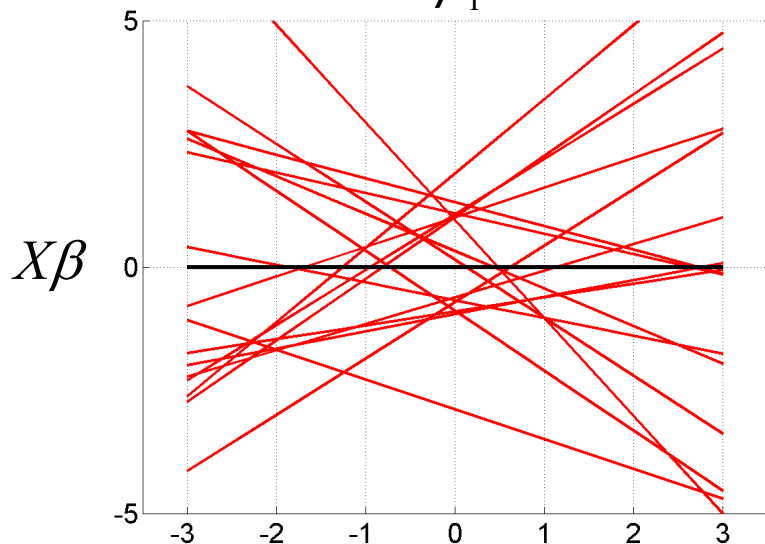


Model:

$$y = X\beta + e$$

Prior:

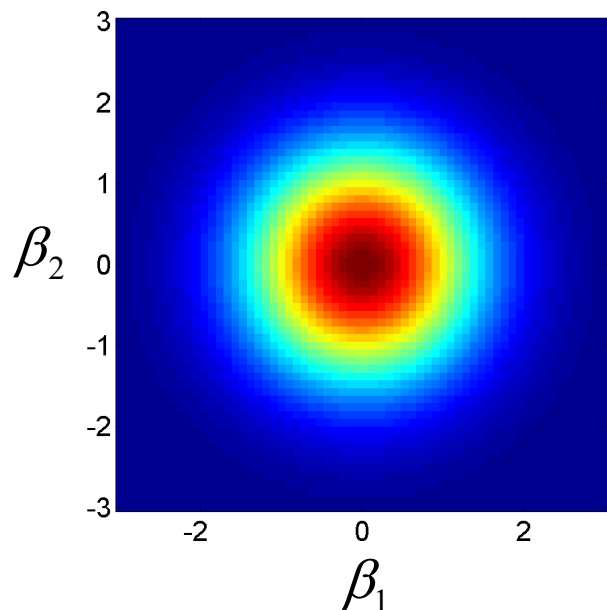
$$p(\beta|\alpha_2) = N_k(0, \alpha_2^{-1} I_k) \\ \propto \exp(-\alpha_2 \|\beta\|^2 / 2)$$



— Sample curves from prior
(before observing any data)

— Mean curve

Bayesian regression: *priors & likelihood*



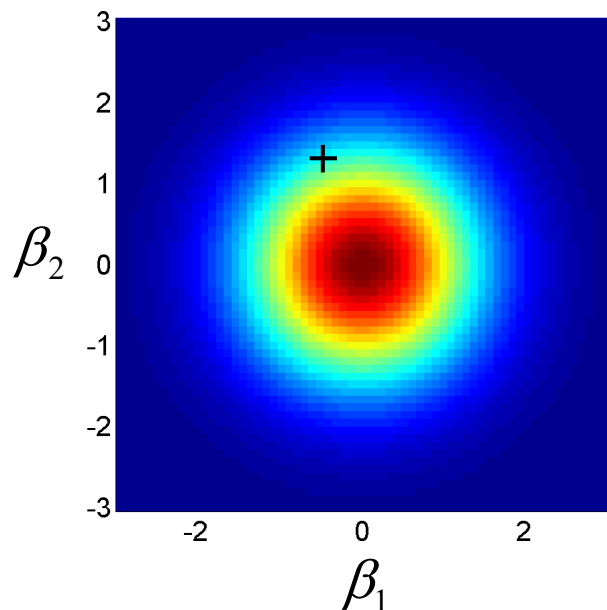
Model: $y = X\beta + e$

Prior: $p(\beta | \alpha_2) = N_k(0, \alpha_2^{-1} I_k)$
 $\propto \exp(-\alpha_2 \|\beta\|^2 / 2)$

Likelihood:

$$p(y | \beta, \alpha_1) = \prod_{i=1}^N p(y_i | \beta, \alpha_1^{-1})$$
$$p(y_i | \beta, \alpha_1) = N(X_i \beta, \alpha_1^{-1})$$
$$\propto \exp(-\alpha_1 (y_i - X_i \beta)^2 / 2)$$

Bayesian regression: *priors & likelihood*



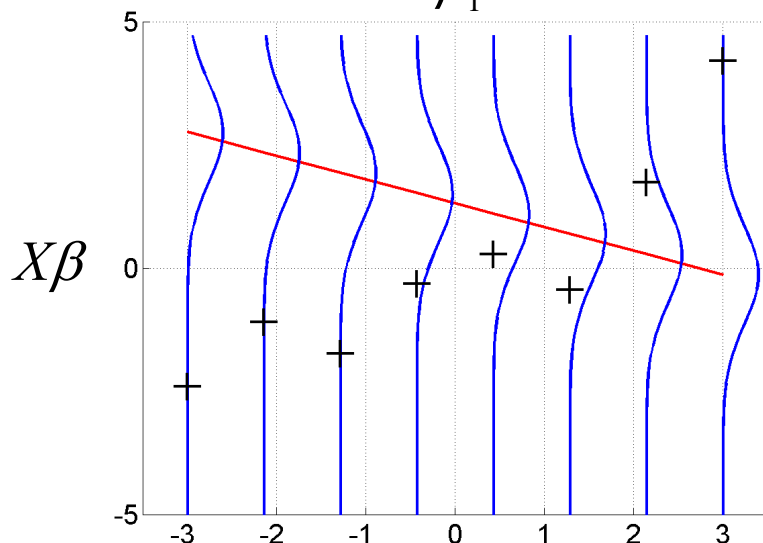
Model: $y = X\beta + e$

Prior: $p(\beta | \alpha_2) = N_k(0, \alpha_2^{-1} I_k)$
 $\propto \exp(-\alpha_2 \|\beta\|^2 / 2)$

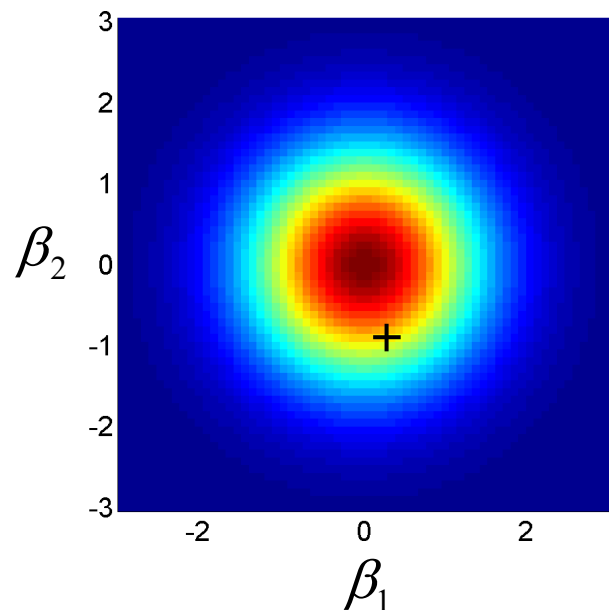
Likelihood:

$$p(y | \beta, \alpha_1) = \prod_{i=1}^N p(y_i | \beta, \alpha_1^{-1})$$

$$p(y_i | \beta, \alpha_1) = N(X_i \beta, \alpha_1^{-1})$$
$$\propto \exp(-\alpha_1 (y_i - X_i \beta)^2 / 2)$$



Bayesian regression: *priors & likelihood*



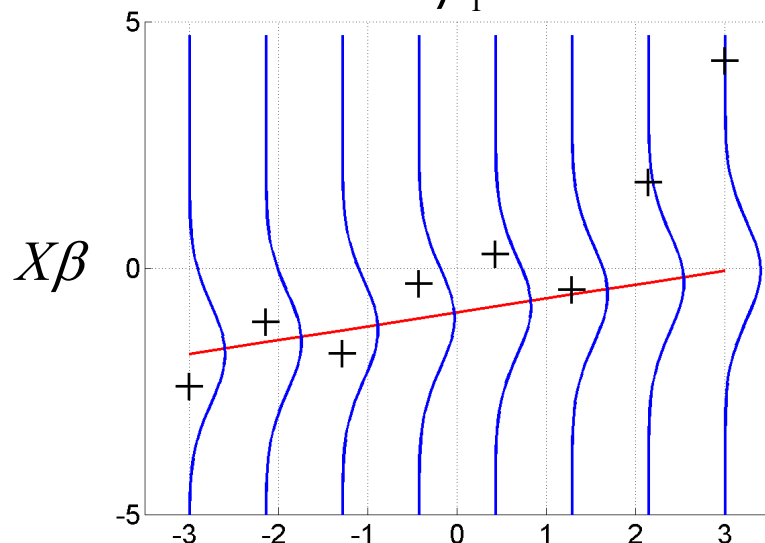
Model: $y = X\beta + e$

Prior: $p(\beta | \alpha_2) = N_k(0, \alpha_2^{-1} I_k)$
 $\propto \exp(-\alpha_2 \|\beta\|^2 / 2)$

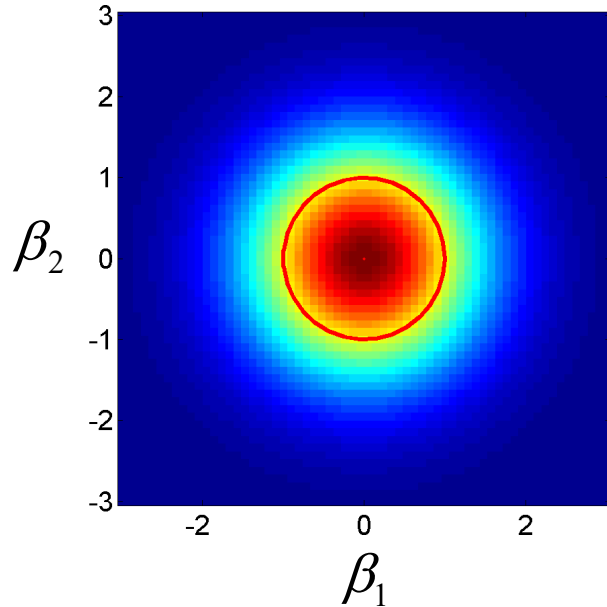
Likelihood:

$$p(y | \beta, \alpha_1) = \prod_{i=1}^N p(y_i | \beta, \alpha_1^{-1})$$

$$p(y_i | \beta, \alpha_1) = N(X_i \beta, \alpha_1^{-1})$$
$$\propto \exp(-\alpha_1 (y_i - X_i \beta)^2 / 2)$$



Bayesian regression: *posterior*



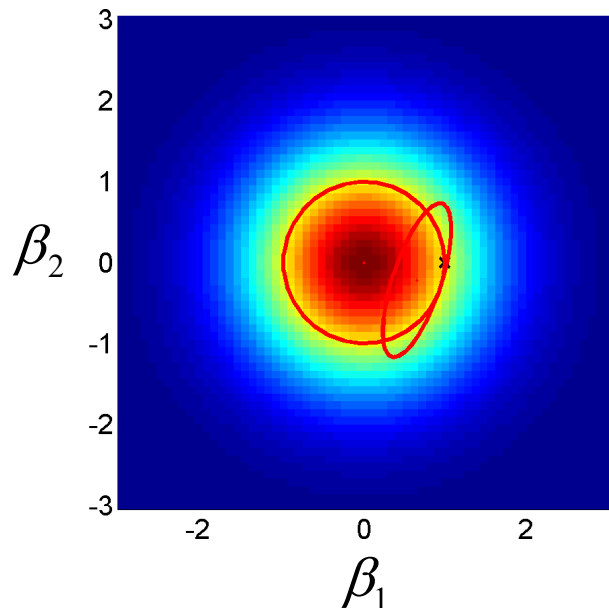
Model: $y = X\beta + e$

Prior: $p(\beta | \alpha_2) = N_k(0, \alpha_2^{-1} I_k)$
 $\propto \exp(-\alpha_2 \|\beta\|^2 / 2)$

Likelihood: $p(y | \beta, \alpha_1) = \prod_{i=1}^N p(y_i | \beta, \alpha_1)$

Bayes Rule: $p(\beta | y, \alpha) \propto p(y | \beta, \alpha) p(\beta | \alpha)$

Bayesian regression: *posterior*



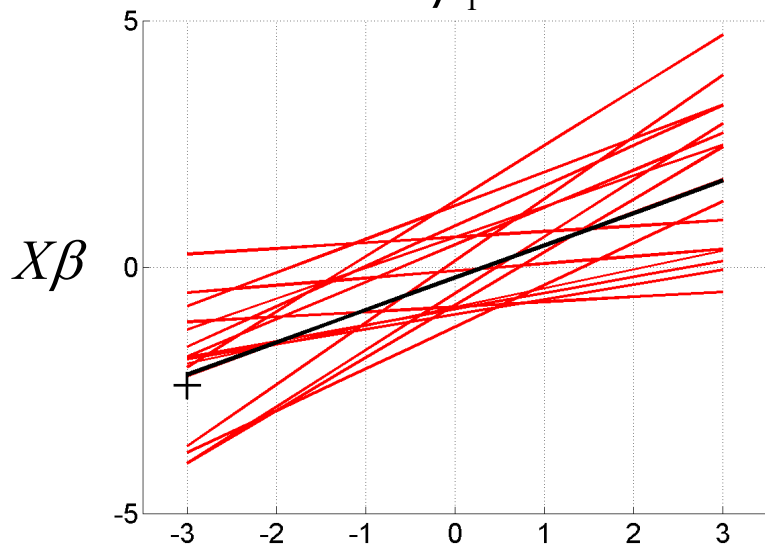
Model: $y = X\beta + e$

Prior: $p(\beta | \alpha_2) = N_k(0, \alpha_2^{-1} I_k)$
 $\propto \exp(-\alpha_2 \|\beta\|^2 / 2)$

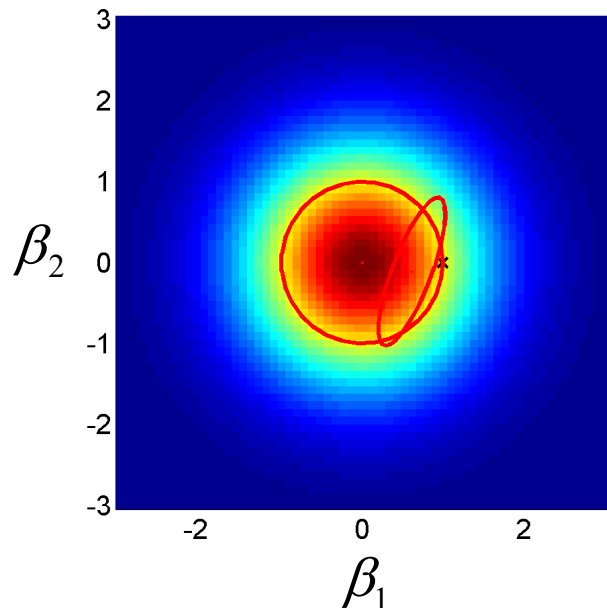
Likelihood: $p(y | \beta, \alpha_1) = \prod_{i=1}^N p(y_i | \beta, \alpha_1)$

Bayes Rule: $p(\beta | y, \alpha) \propto p(y | \beta, \alpha) p(\beta | \alpha)$

Posterior: $p(\beta | y, \alpha) = N(\mu, C)$
 $C = (\alpha_1 X^T X + \alpha_2 I_k)^{-1}$
 $\mu = \alpha_1 C X^T y$



Bayesian regression: *posterior*



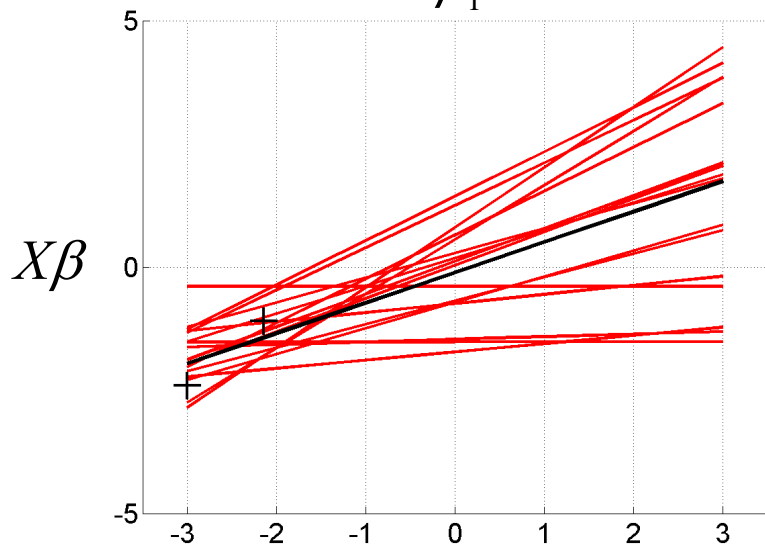
Model: $y = X\beta + e$

Prior: $p(\beta | \alpha_2) = N_k(0, \alpha_2^{-1} I_k)$
 $\propto \exp(-\alpha_2 \|\beta\|^2 / 2)$

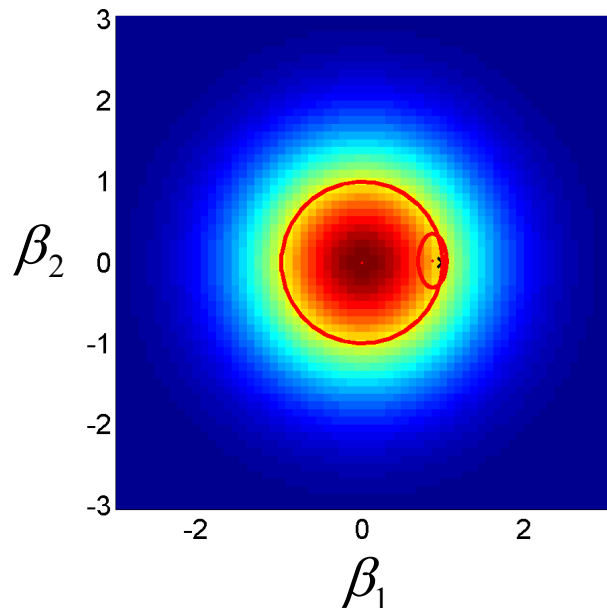
Likelihood: $p(y | \beta, \alpha_1) = \prod_{i=1}^N p(y_i | \beta, \alpha_1)$

Bayes Rule: $p(\beta | y, \alpha) \propto p(y | \beta, \alpha) p(\beta | \alpha)$

Posterior: $p(\beta | y, \alpha) = N(\mu, C)$
 $C = (\alpha_1 X^T X + \alpha_2 I_k)^{-1}$
 $\mu = \alpha_1 C X^T y$



Bayesian regression: *posterior*



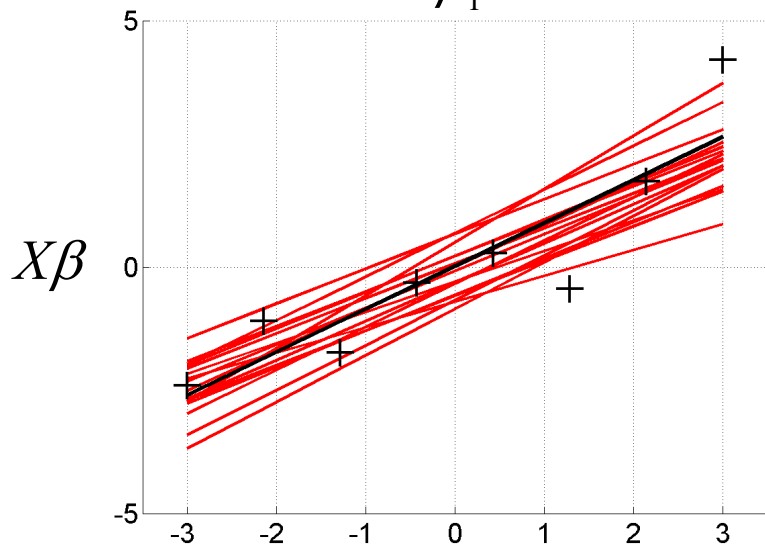
Model: $y = X\beta + e$

Prior: $p(\beta | \alpha_2) = N_k(0, \alpha_2^{-1} I_k)$
 $\propto \exp(-\alpha_2 \|\beta\|^2 / 2)$

Likelihood: $p(y | \beta, \alpha_1) = \prod_{i=1}^N p(y_i | \beta, \alpha_1)$

Bayes Rule: $p(\beta | y, \alpha) \propto p(y | \beta, \alpha) p(\beta | \alpha)$

Posterior: $p(\beta | y, \alpha) = N(\mu, C)$
 $C = (\alpha_1 X^T X + \alpha_2 I_k)^{-1}$
 $\mu = \alpha_1 C X^T y$



Question

$$p(\beta \mid y, \alpha) = N(\mu, C)$$

$$C = (\alpha_1 X^T X + \alpha_2 I_k)^{-1}$$

$$\mu = \alpha_1 C X^T y$$

- What is the point estimate for the model parameter beta from the above?

Maximum A Posteriori

Week 6 Contents / Objectives

Part A

- Bayesian Inference
- Bayesian Linear Regression
- **Predictive Distribution**

Part B

- Computational Graph
- PyTorch: A Deep Learning Library

MAP vs MLE

- MAP: Maximum A Posteriori

$$\beta^* = \mu = \alpha_1 [\alpha_1 \mathbf{X}^\top \mathbf{X} + \alpha_2 \mathbf{I}_k]^{-1} \mathbf{X}^\top \mathbf{y}$$

$$\beta^* = \mu = \left[\mathbf{X}^\top \mathbf{X} + \frac{\alpha_2}{\alpha_1} \mathbf{I}_k \right]^{-1} \mathbf{X}^\top \mathbf{y}$$

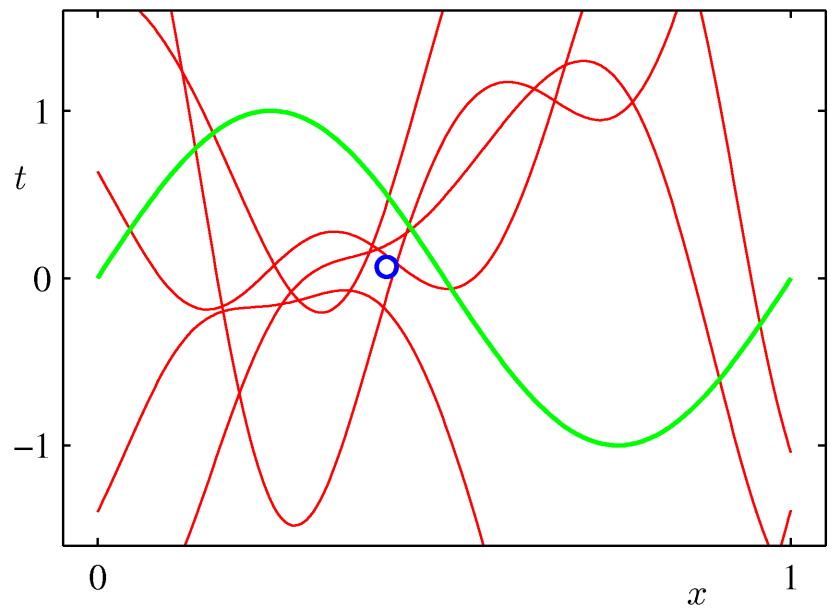
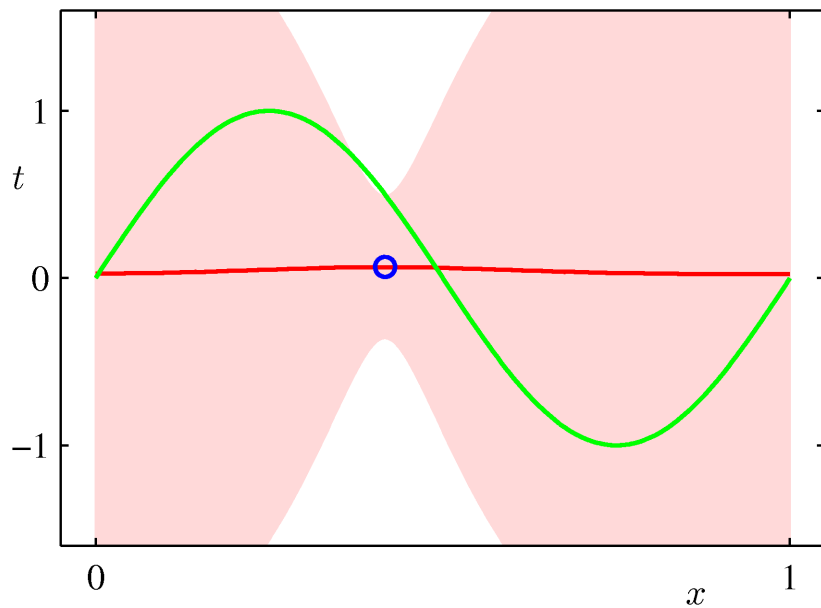
- MLE: Maximum Likelihood Estimation

$$\beta^* = [\mathbf{X}^\top \mathbf{X} + \mathbf{I}_k]^{-1} \mathbf{X}^\top \mathbf{y}$$

- MAP = regularised MLE

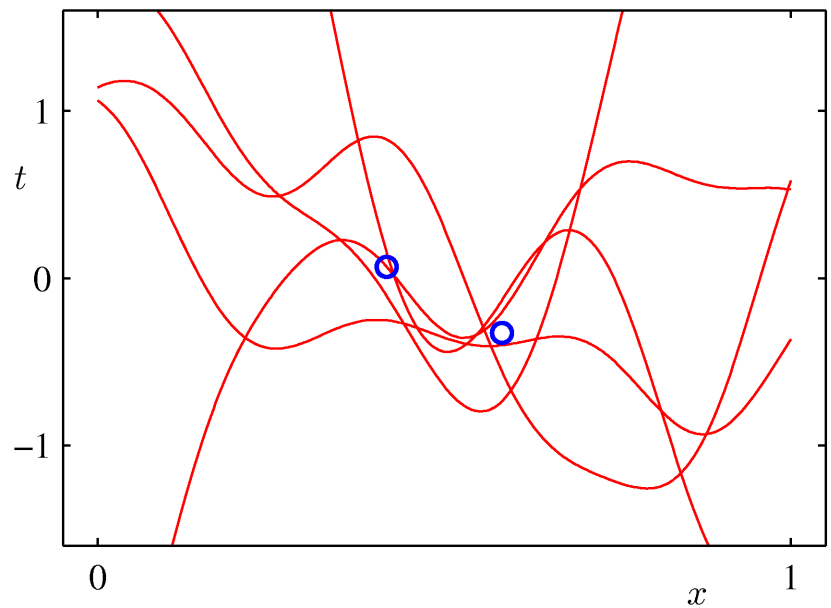
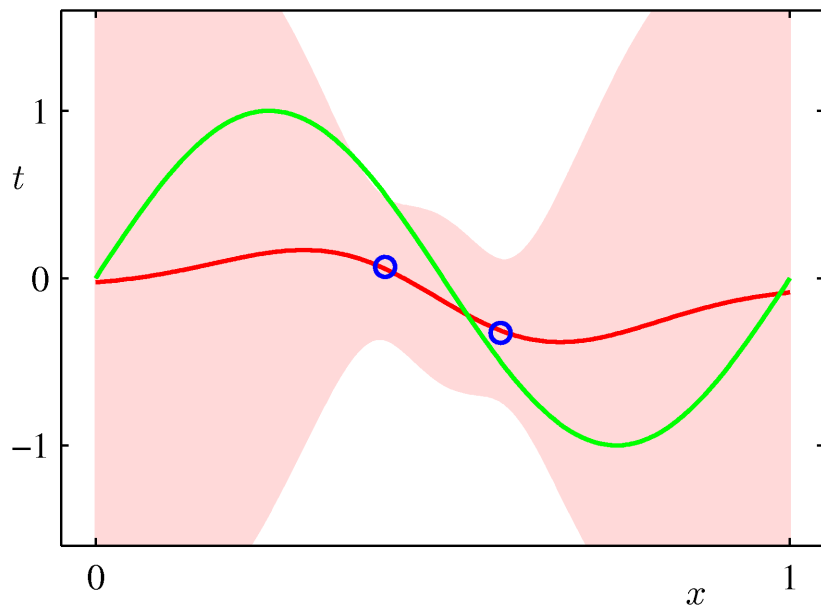
Predictive Distribution (1)

- Example: Sinusoidal data, 9 Gaussian basis functions, 1 data point



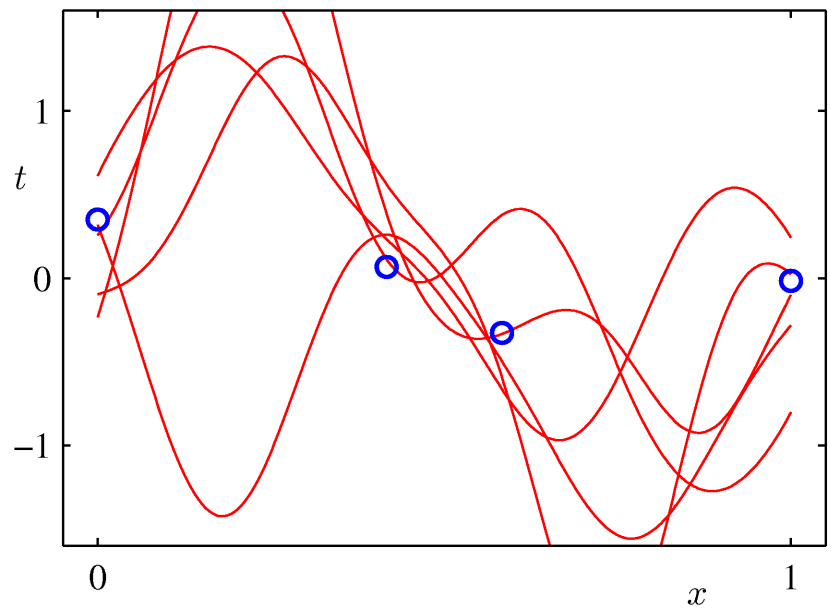
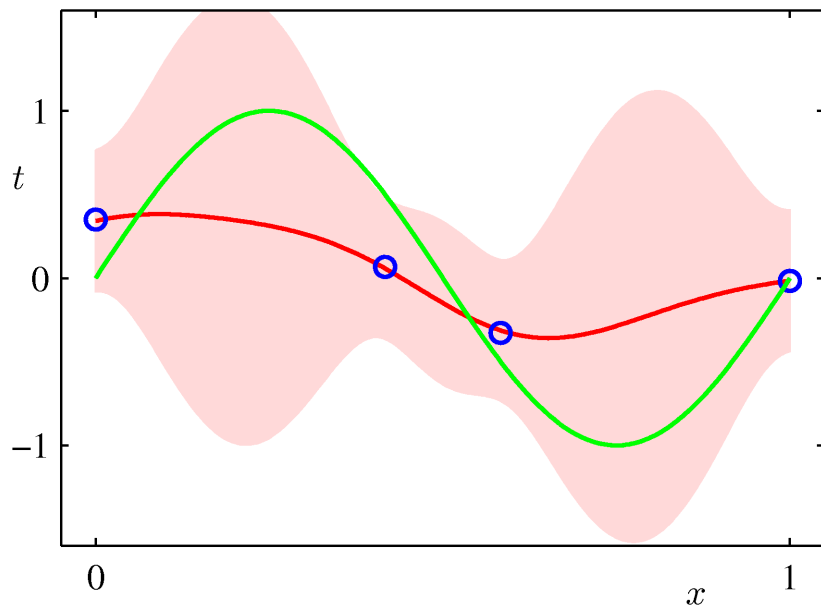
Predictive Distribution (2)

- Example: Sinusoidal data, 9 Gaussian basis functions, 2 data points



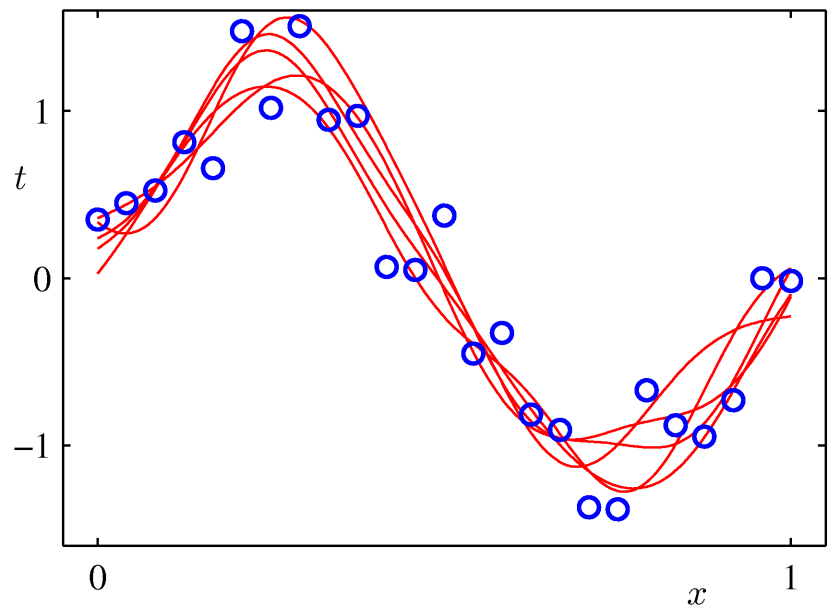
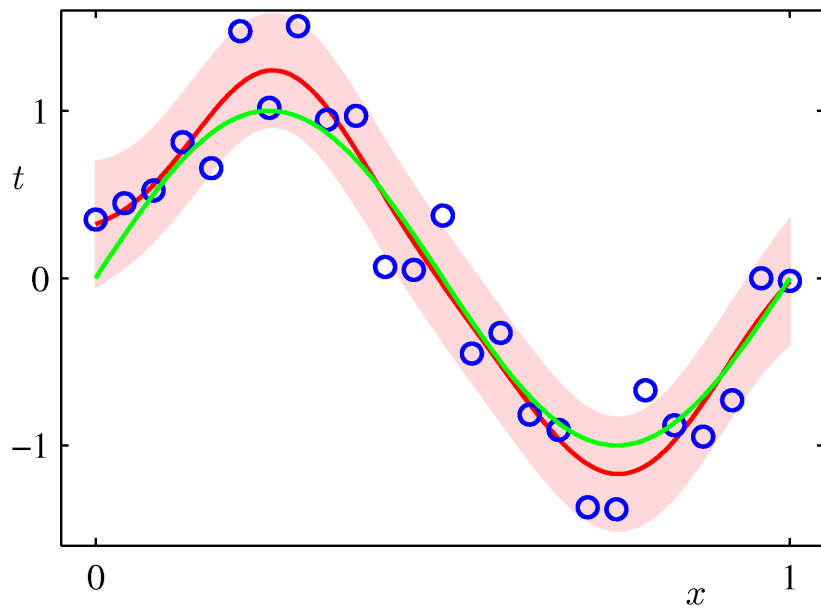
Predictive Distribution (3)

- Example: Sinusoidal data, 9 Gaussian basis functions, 4 data points



Predictive Distribution (4)

- Example: Sinusoidal data, 9 Gaussian basis functions, 25 data points



- Notice the decreased uncertainty

Pros and Cons of Bayesian

- Advantages
 - Deal with uncertainty.
 - Make use of more information (prior, if available)
 - Less overfitting in general
- Disadvantages
 - Complexity
 - Subjectivity: all inferences are based on beliefs. Which prior to choose?

Summary on Bayesian regression

- Bayesian inference: placing a probability distribution (prior density) over the model parameters for their probabilistic interpretation
- Bayesian regression: Bayesian treatment of linear regression (model parameters)
- Key Bayesian concepts: prior, posterior, likelihood, marginal likelihood, Bayes' rule, marginalisation

Week 6 Contents / Objectives

Part A

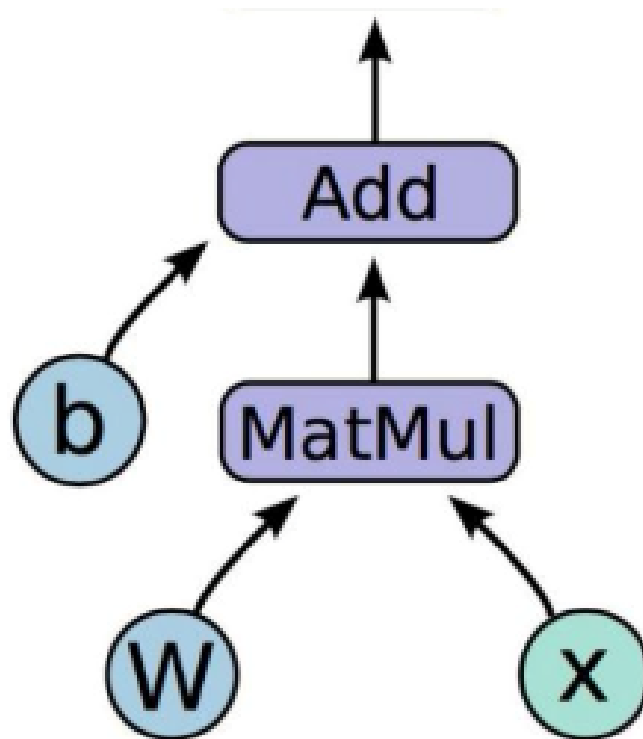
- Bayesian Inference
- Bayesian Linear Regression
- Predictive Distribution

Part B

- **Computational Graph**
- PyTorch: A Deep Learning Library

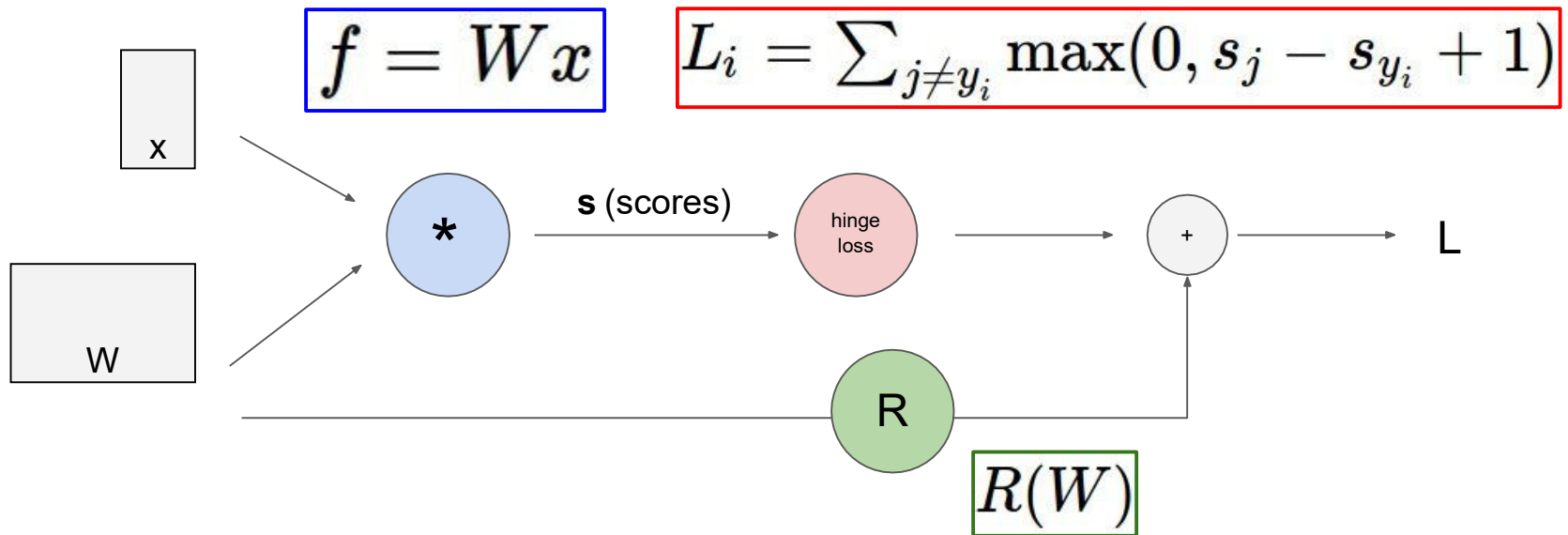
Computational Graph: L Regression

- Linear regression $y = \mathbf{W}\mathbf{x} + \mathbf{b}$

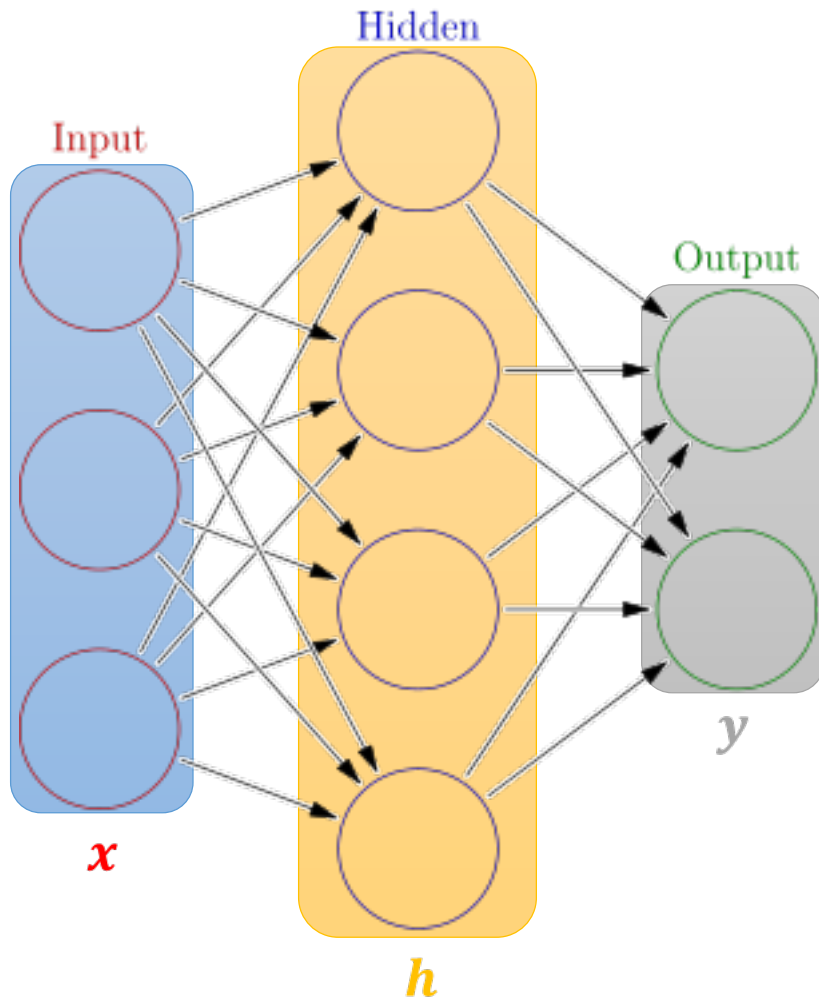


Source: Nelson Liu: <https://colab.research.google.com/drive/1iLtGFDpnIuHj5B0rQDGG5lqq6BQ8FRh>

Computational Graph: w/t Reg.



Multilayer Perceptron (NN) vs LR



$$h = \sigma(W_1 x + b_1)$$
$$y = \sigma(W_2 h + b_2)$$

Weights

Activation functions



Question:
How many model
parameters?

$$[3 \times 4] + [4 \times 2] = 20 \text{ weights}$$

$$4 + 2 = 6 \text{ biases}$$

26 learnable parameters

4 + 2 = 6 neurons (not counting inputs)

THAT'S NOT ENOUGH



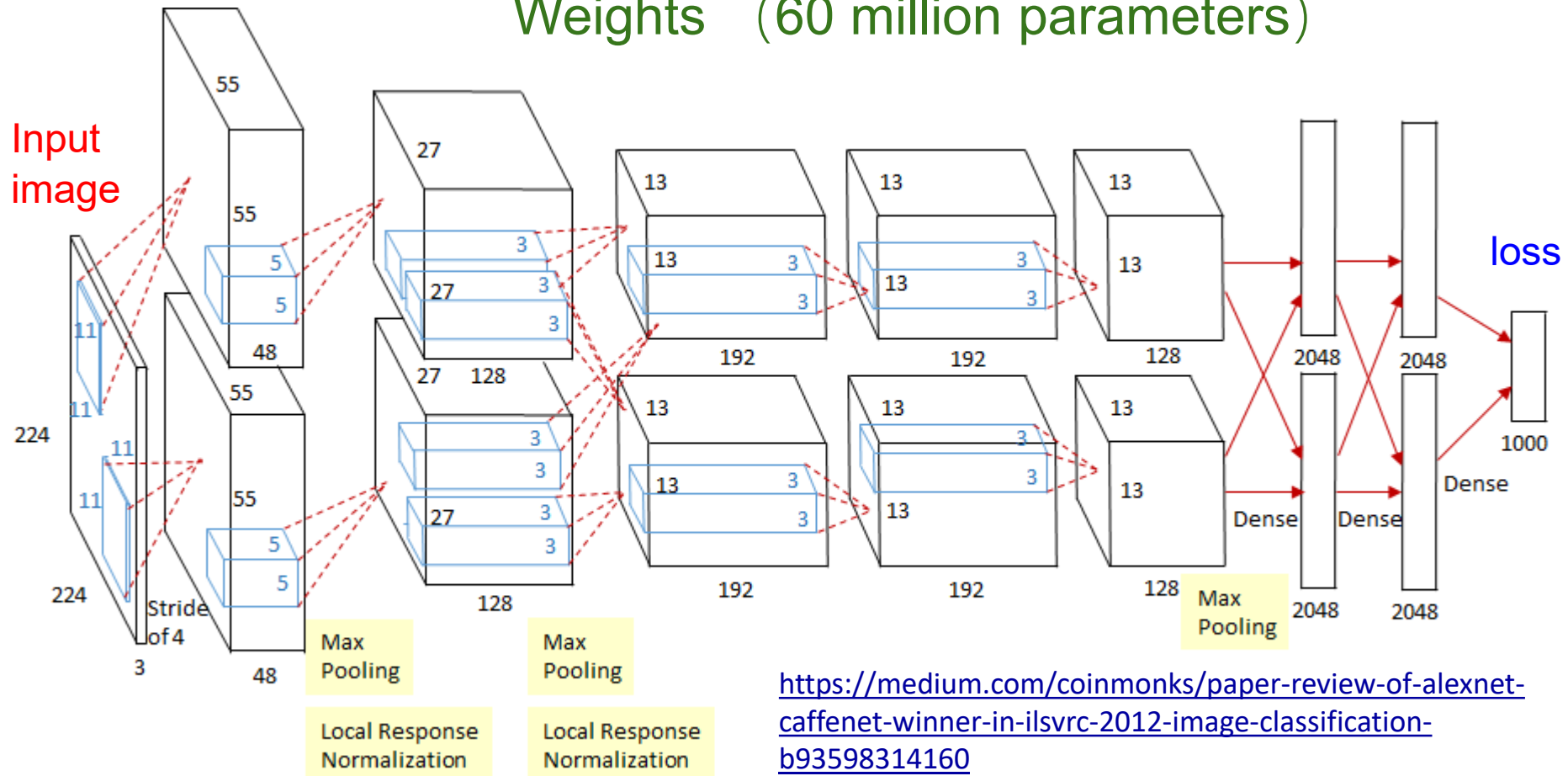
WE HAVE TO GO DEEPER

Source: https://miro.medium.com/max/570/0*y8AuUHSoTGRqX40h.jpeg

quickmeme.com

Computational Graph: DL

Weights (60 million parameters)



Week 6 Contents / Objectives

Part A

- Bayesian Inference
- Bayesian Linear Regression
- Predictive Distribution

Part B

- Computational Graph
- **PyTorch: A Deep Learning Library**

PyTorch



- An open source deep learning library by Facebook
 - **Tensor** computing (like NumPy) with strong acceleration via graphics processing units (GPU)
 - Deep neural networks built on a tape-based autodiff system
- **torch.Tensor**
 - multidimensional array data structures (arrays) for programming
 - Scalar: 0D tensor
 - Vector: 1D tensor
 - Matrix: 2D tensor

Key Modules in PyTorch

- **torch.autograd**

- Automatic differentiation. A recorder records what operations have performed, and then it replays it backward to compute the gradients.

- **torch.optim**

- Implementation of various optimization algorithms used for building neural networks (and other ML algorithms).

- **torch.nn**

- High-level definition of the **computational graphs** of complex neural networks (and other ML algorithms)

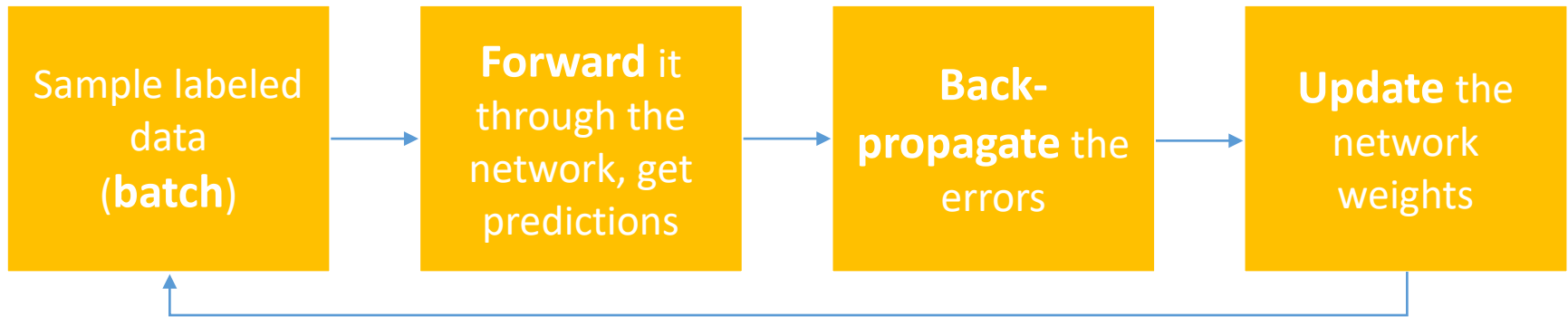
Dynamic Computational Graph

A graph is created on the fly

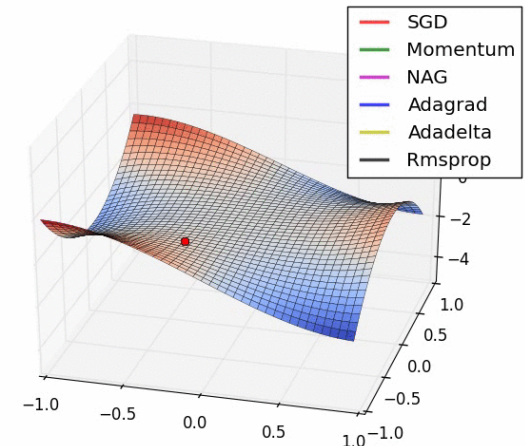
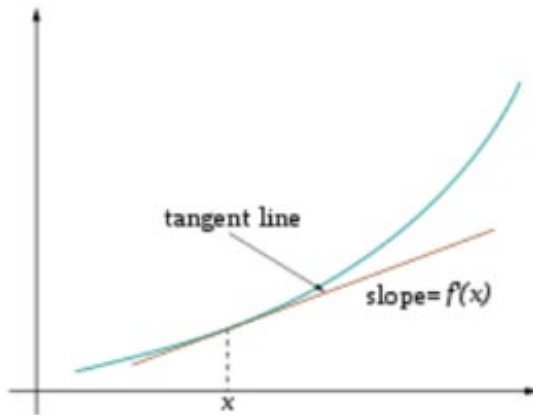
```
W_h = torch.randn(20, 20, requires_grad=True)
W_x = torch.randn(20, 10, requires_grad=True)
x = torch.randn(1, 10)
prev_h = torch.randn(1, 20)
```



Training



Optimize (min. or max.) **objective/cost function $J(\theta)$**
Generate **error signal** that measures difference between predictions and target values



Use error signal to change the **weights** and get more accurate predictions
Subtracting a fraction of the **gradient** moves you towards the **(local) minimum of the cost function**

<https://medium.com/@ramrajchandradevan/the-evolution-of-gradient-descend-optimization-algorithm-4106a6702d39>

Acknowledgement

- Part A used materials from: Christopher Bishop, Neil Lawrence, Lee Harrison, John Gosling, Chuck Huber
- Part B used materials from: Ismini Lourentzou, Fei-Fei Li & Justin Johnson & Serena Yeung, Rui Zhang, Nelson Liu

Recommended Reading

- PRML book: Section 3.3 on Bayesian Linear Regression
- The lab notebook and references there