Linear Regression

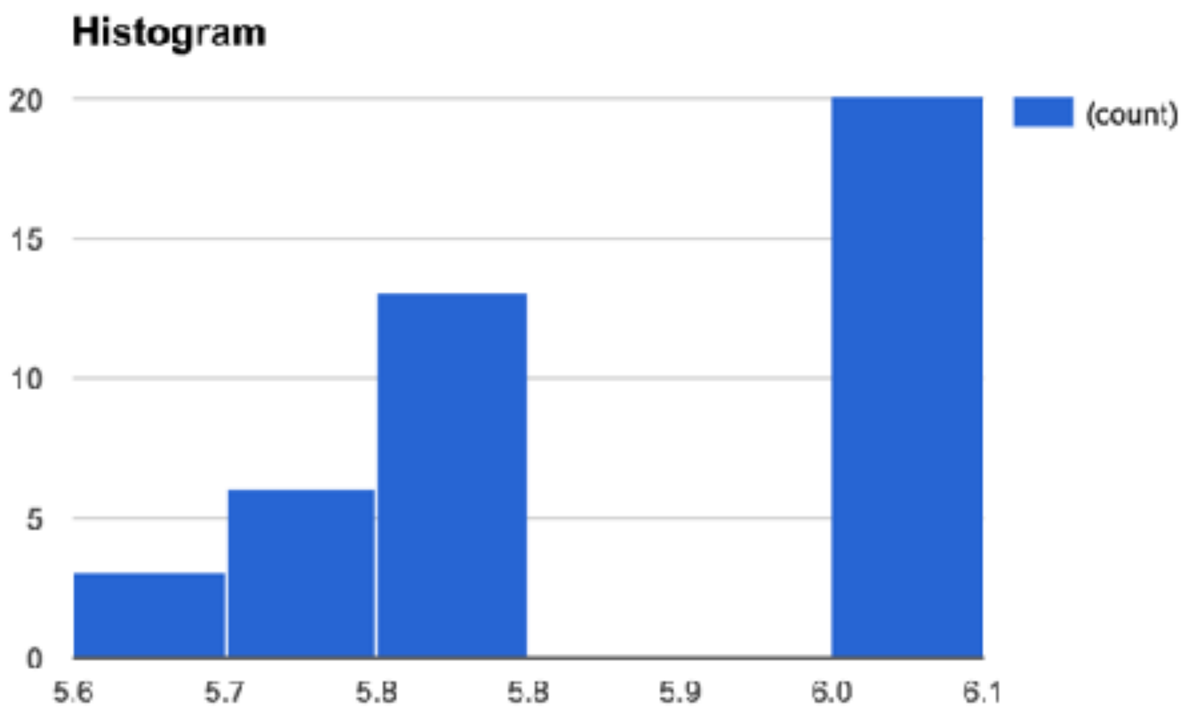
Instructor: Alan Ritter

Many Slides from Tom Mitchell

TA

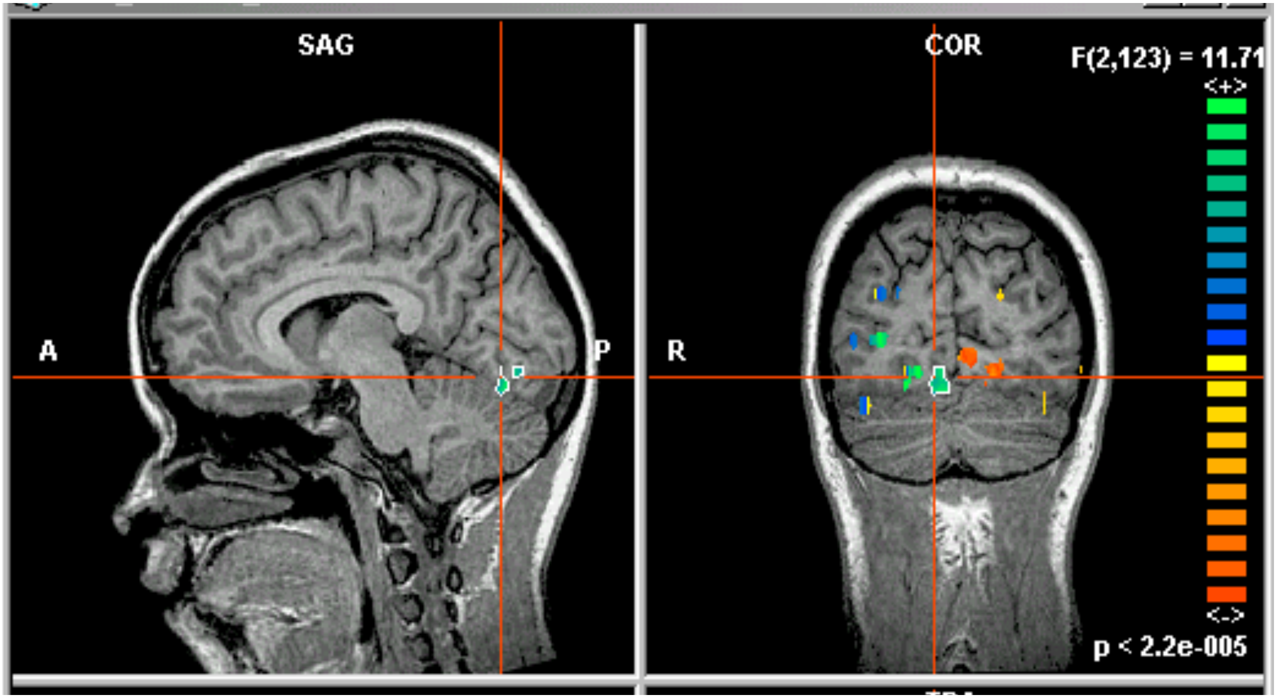
- Chaoyue Liu
  - liu.2656@buckeyemail.osu.edu
  - DL 586

HW1



What if we have continuous X_i ?

Eg., image classification: X_i is real-valued i[th] pixel



What if we have continuous X_i ?

Eg., image classification: X_i is real-valued ith pixel

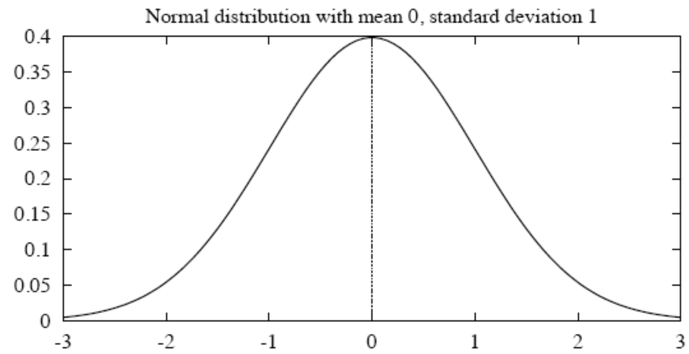
Naïve Bayes requires $P(X_i \mid Y=y_k)$, but X_i is real (continuous)

$$P(Y = y_k|X_1 \ldots X_n) = \frac{P(Y = y_k) \prod_i P(X_i|Y = y_k)}{\sum_j P(Y = y_j) \prod_i P(X_i|Y = y_j)}$$

Common approach: assume $P(X_i \mid Y=y_k)$ follows a Normal (Gaussian) distribution

Gaussian Distribution
(also called "Normal")

p(x) is a *probability density function*, whose integral (not sum) is 1



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

The probability that X will fall into the interval (a, b) is given by

$$\int_a^b p(x)dx$$

- Expected, or mean value of X, $E[X]$, is

$$E[X] = \mu$$

- Variance of X is

$$Var(X) = \sigma^2$$

- Standard deviation of X, σ_X, is

$$\sigma_X = \sigma$$

What if we have continuous X_i?

Gaussian Naïve Bayes (GNB): assume

$$p(X_i = x | Y = y_k) = \frac{1}{\sqrt{2\pi\sigma_{ik}^2}} \; e^{-\frac{1}{2}\left(\frac{x-\mu_{ik}}{\sigma_{ik}}\right)^2}$$

Sometimes assume variance

- is independent of Y (i.e., σ_i),
- or independent of X_i (i.e., σ_k)
- or both (i.e., σ)

Gaussian Naïve Bayes Algorithm – continuous X_i (but still discrete Y)

- Train Naïve Bayes (examples)

  for each value y_k

  estimate* $\pi_k \equiv P(Y = y_k)$

  for each attribute X_i estimate $P(X_i|Y = y_k)$

  - class conditional mean μ_{ik}, variance σ_{ik}

- Classify (X^{new})

$$Y^{new} \leftarrow \arg\max_{y_k} \; P(Y = y_k) \prod_i P(X_i^{new}|Y = y_k)$$

$$Y^{new} \leftarrow \arg\max_{y_k} \; \pi_k \prod_i \mathcal{N}(X_i^{new}; \mu_{ik}, \sigma_{ik})$$

* probabilities must sum to 1, so need estimate only n-1 parameters...

Estimating Parameters: Y discrete, X_i continuous

Maximum likelihood estimates:

jth training example

$$\widehat{\mu}_{ik} = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j X_i^j \delta(Y^j = y_k)$$

ith feature

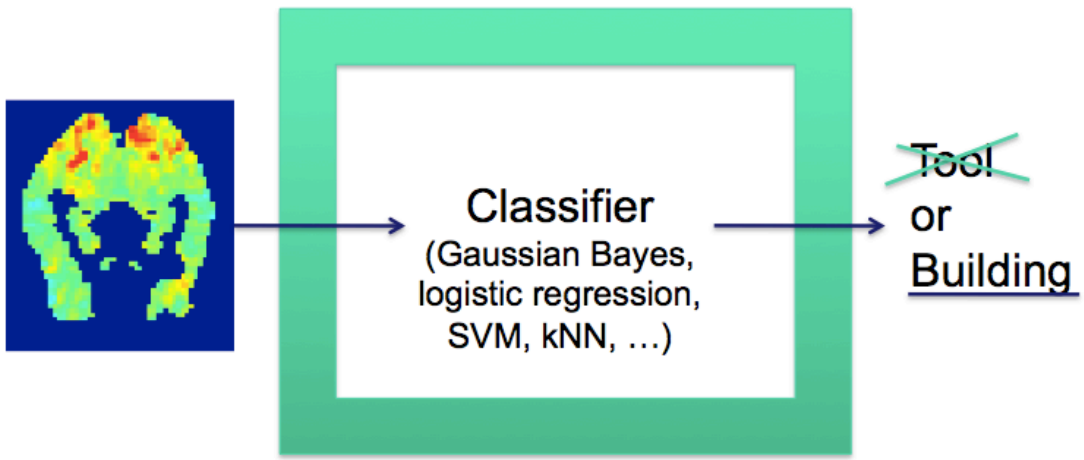
kth class

δ()=1 if ($Y^j=y_k$) else 0

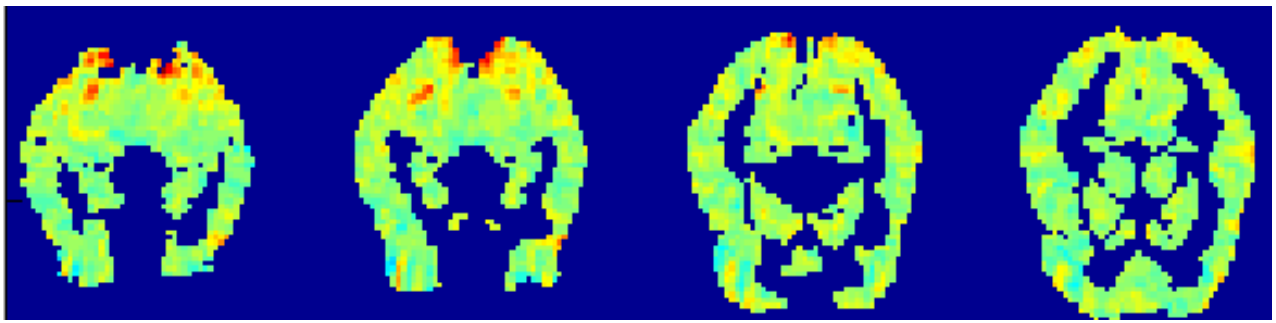
$$\widehat{\sigma}_{ik}^2 = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j (X_i^j - \widehat{\mu}_{ik})^2 \delta(Y^j = y_k)$$

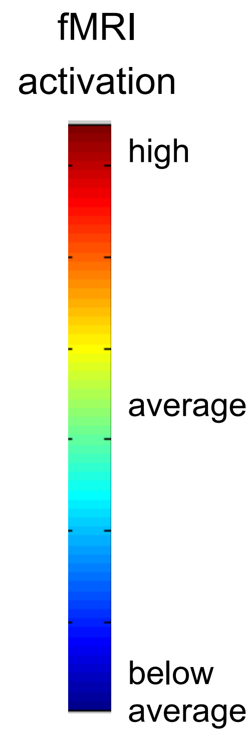
GNB Example: Classify a person's cognitive state, based on brain image

- reading a sentence or viewing a picture?
- reading the word describing a "Tool" or "Building"?
- answering the question, or getting confused?



Mean activations over all training examples for Y="bottle"

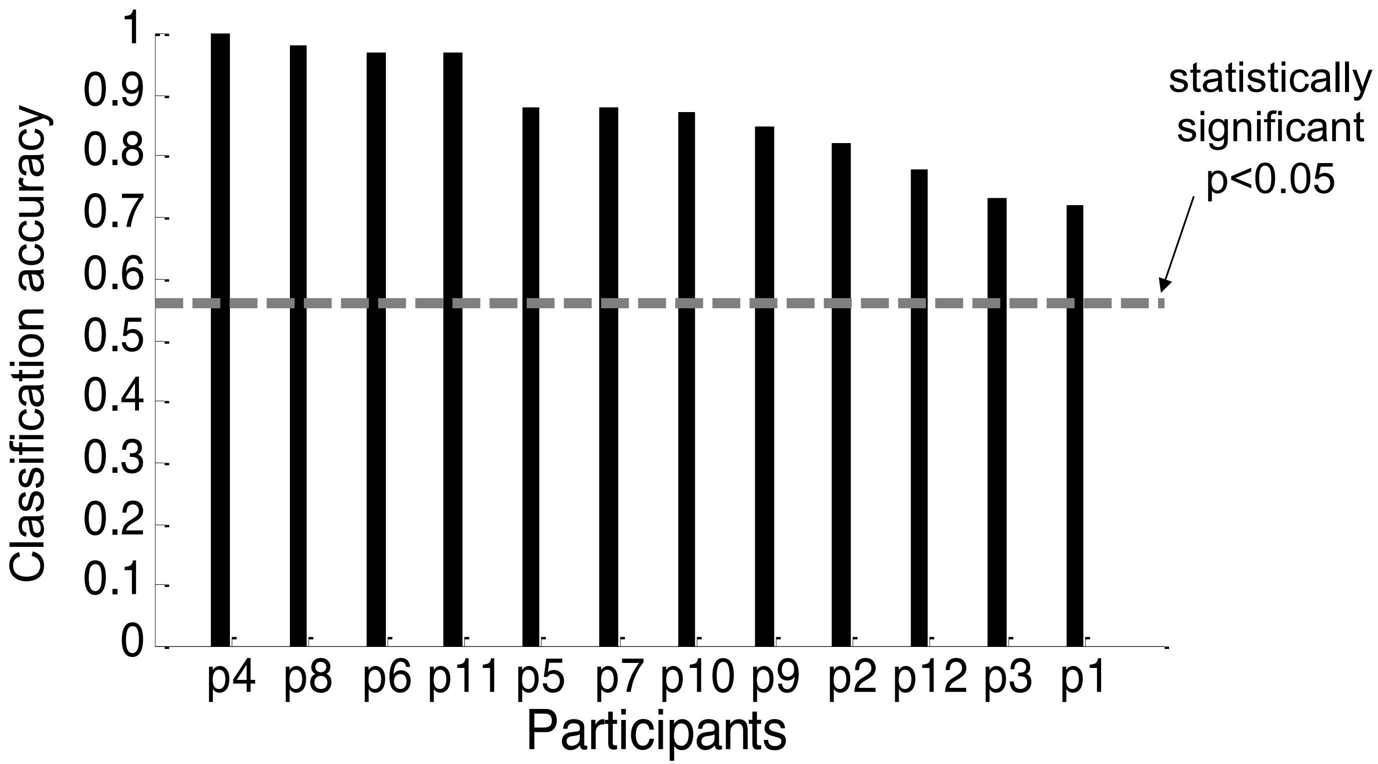




Y is the mental state (reading "house" or "bottle")
X_i are the voxel activities,

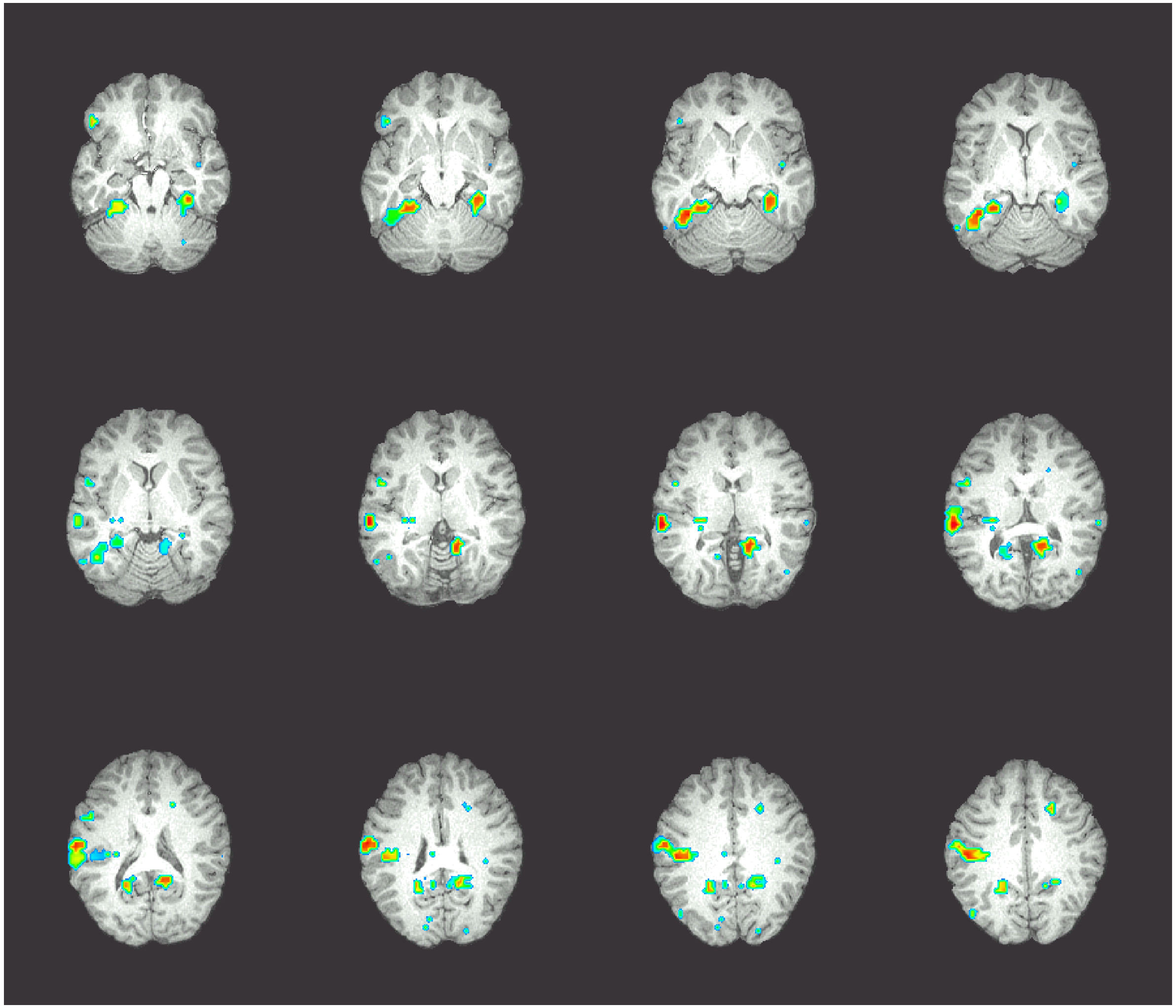
this is a plot of the μ's defining $P(X_i \mid Y=\text{"bottle"})$

Classification task: is person viewing a “tool” or “building”?



Where is information encoded in the brain?

Accuracies of cubical 27-voxel classifiers centered at each significant voxel [0.7-0.8]



Naïve Bayes: What you should know

- Designing classifiers based on Bayes rule
- Conditional independence
  - What it is
  - Why it's important
- Naïve Bayes assumption and its consequences
  - Which (and how many) parameters must be estimated under different generative models (different forms for P(X|Y))
    - and why this matters
- How to train Naïve Bayes classifiers
  - MLE and MAP estimates
  - with discrete and/or continuous inputs X_i

Regression

So far, we've been interested in learning $P(Y|X)$ where Y has discrete values (called 'classification')

What if Y is continuous? (called 'regression')

- predict weight from gender, height, age, …
- predict Google stock price today from Google, Yahoo, MSFT prices yesterday
- predict each pixel intensity in robot's current camera image, from previous image and previous action

Regression

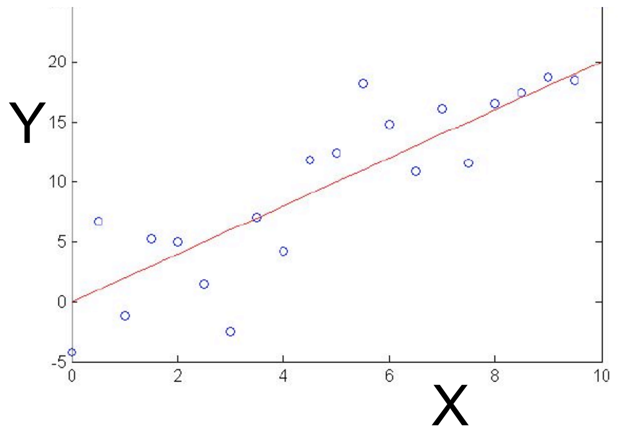
Wish to learn $f: X \rightarrow Y$, where Y is real, given $\{<x^1,y^1>\ldots<x^n,y^n>\}$

Approach:

1. choose some parameterized form for $P(Y|X; \theta)$
   (θ is the vector of parameters)

2. derive learning algorithm as MCLE or MAP estimate for θ

1. Choose parameterized form for P(Y|X; θ)



Assume Y is some deterministic f(X), plus random noise

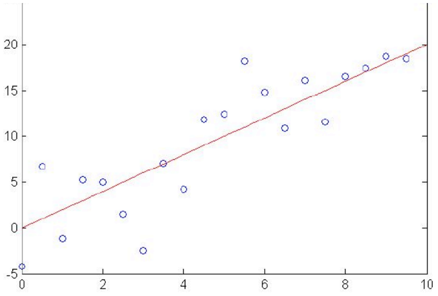
$$y = f(x) + \epsilon \qquad \text{where} \quad \epsilon \sim N(0, \sigma)$$

Therefore Y is a random variable that follows the distribution

$$p(y|x) = N(f(x), \sigma)$$

and the expected value of y for any given x is f(x)

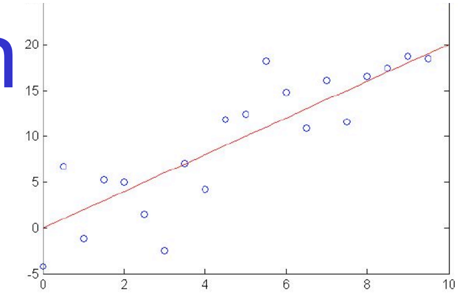
Training Linear Regression



$$p(y|x;W) = N(w_0 + w_1 x, \sigma)$$

How can we learn W from the training data?

Training Linear Regression



$$p(y|x;W) = N(w_0 + w_1 x, \sigma)$$

How can we learn W from the training data?

Learn Maximum Conditional Likelihood Estimate!

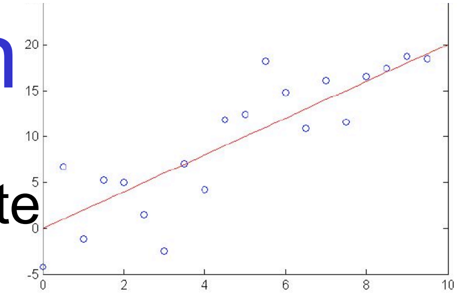
$$W_{MCLE} = \arg\max_W \prod_l p(y^l | x^l, W)$$

$$W_{MCLE} = \arg\max_W \sum_l \ln p(y^l | x^l, W)$$

where

$$p(y|x;W) = \frac{1}{\sqrt{2\pi\sigma^2}} \; e^{-\frac{1}{2}\left(\frac{y - f(x;W)}{\sigma}\right)^2}$$

Training Linear Regression

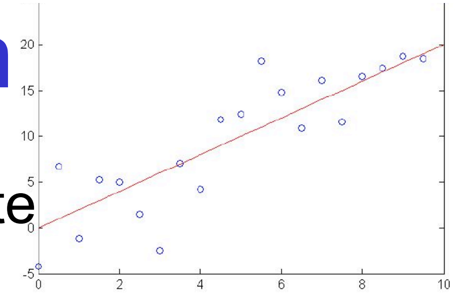


Learn Maximum Conditional Likelihood Estimate

$$W_{MCLE} = \arg\max_{W} \sum_{l} \ln p(y^l | x^l, W)$$

where $$p(y|x;W) = \frac{1}{\sqrt{2\pi\sigma^2}} \; e^{-\frac{1}{2}\left(\frac{y - f(x;W)}{\sigma}\right)^2}$$

Training Linear Regression



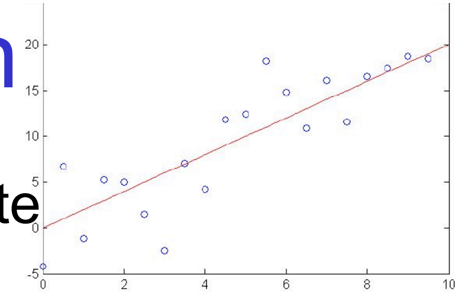
Learn Maximum Conditional Likelihood Estimate

$$W_{MCLE} = \arg\max_{W} \sum_{l} \ln p(y^l | x^l, W)$$

where
$$p(y|x;W) = \frac{1}{\sqrt{2\pi\sigma^2}} \; e^{-\frac{1}{2}\left(\frac{y-f(x;W)}{\sigma}\right)^2}$$

so:
$$W_{MCLE} = \arg\min_{W} \sum_{l} (y - f(x;W))^2$$

Training Linear Regression



Learn Maximum Conditional Likelihood Estimate

$$W_{MCLE} = \arg\min_{W} \sum_{l} (y - f(x; W))^2$$

Can we derive gradient descent rule for training?

$$\frac{\partial \sum_l (y - f(x;W))^2}{\partial w_i} = \sum_l 2(y - f(x;W)) \frac{\partial (y - f(x;W))}{\partial w_i}$$

$$= \sum_l -2(y - f(x;W)) \frac{\partial f(x;W)}{\partial w_i}$$

How about MAP instead of MLE estimate?

$$W = \arg\max_W \ln N(W|0,I) + \sum_l \ln(P(Y^l|X^l;W))$$

$$= \arg\max_W \; c\sum_i w_i^2 + \sum_l \ln(P(Y^l|X^l;W))$$

Regression – What you should know

Under general assumption $p(y|x;W) = N(f(x;W), \sigma)$

1. MLE corresponds to minimizing sum of squared prediction errors

2. MAP estimate minimizes SSE plus sum of squared weights

3. Again, learning is an optimization problem once we choose our objective function
   - maximize data likelihood
   - maximize posterior prob of W

4. Again, we can use gradient descent as a general learning algorithm
   - as long as our objective fn is differentiable wrt W
   - though we might learn local optima ins

5. Almost nothing we said here required that f(x) be linear in x