Machine Learning 10-601

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University

October 6, 2011

Today:
- Linear regression
- Bias/Variance/Unavoidable errors
- Bayes Nets

Readings:

Required:
- Bishop: Chapt. 3 through 3.2
- Bishop: Chapt. 8 through 8.2

Regression

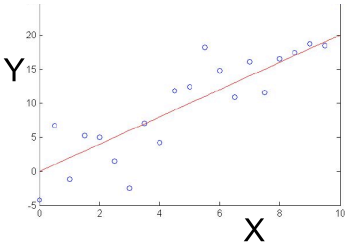
Wish to learn f:X→Y, where Y is real, given $\{<x^1,y^1>\ldots<x^n,y^n>\}$

Approach:

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

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

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



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

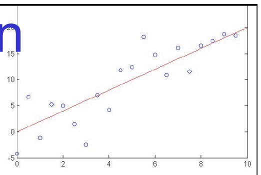
$$y = f(x) + \epsilon \qquad \text{where} \quad \epsilon \sim N(0, \sigma)$$

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 $E_{p(x,y)}[y]=f(x)$

Consider Linear Regression



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

E.g., assume f(x) is linear function of x

$$f(x) = w_0 + \sum w_i x_i$$

$$p(y|x) = N(w_0 + \sum_i w_i x_i, \sigma)$$

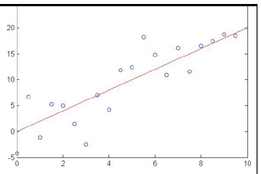
$$E_{p(x,y)}[y|x] = w_0 + \sum_i w_i x_i$$

Notation: to make our parameters explicit, let's write

$$W =< w_0, w_1 \ldots w_n >$$

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

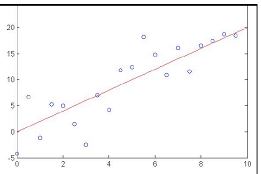
Training Linear Regression



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

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!

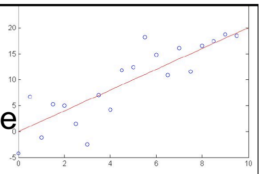
$$W_{MCLE} = \arg\max_W \prod_l p(y^l | x^l, W)$$

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

where

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

Training and Regression



Learn Maximum Conditional Likelihood Estimate

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

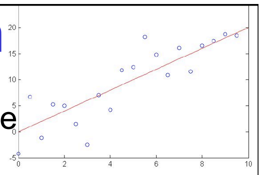
where $$p(y|x;W) = \frac{1}{\sqrt{2\pi\sigma^2}} \; e^{-\frac{1}{2}(\frac{y-f(x;W)}{\sigma})^2}$$

$$\sum_l \ln p(y^l | x^l; W) =$$

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

$$= \arg\min_W \sum_l (y^l - f(x^l;W))^2$$

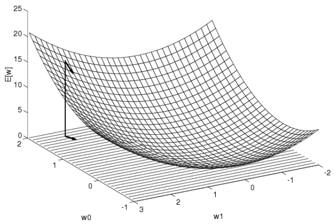
Training Linear Regression



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?



Gradient

$$\nabla E[\vec{w}] \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \cdots \frac{\partial E}{\partial w_n}\right]$$

Training rule:

$$\Delta \vec{w} = -\eta \nabla E[\vec{w}]$$

i.e.,

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

Gradient Descent:

Batch gradient: use error $E_D(\mathbf{w})$ over entire training set D

Do until satisfied:

1. Compute the gradient $\nabla E_D(\mathbf{w}) = \left[\frac{\partial E_D(\mathbf{w})}{\partial w_0} \ldots \frac{\partial E_D(\mathbf{w})}{\partial w_n}\right]$
2. Update the vector of parameters: $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla E_D(\mathbf{w})$

Stochastic gradient: use error $E_d(\mathbf{w})$ over single examples $d \in D$

Do until satisfied:

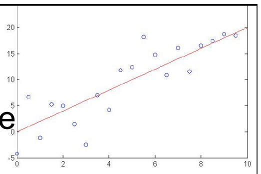
1. Choose (with replacement) a random training example $d \in D$
2. Compute the gradient just for d: $\nabla E_d(\mathbf{w}) = \left[\frac{\partial E_d(\mathbf{w})}{\partial w_0} \ldots \frac{\partial E_d(\mathbf{w})}{\partial w_n}\right]$
3. Update the vector of parameters: $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla E_d(\mathbf{w})$

Stochastic approximates Batch arbitrarily closely as $\eta \rightarrow 0$

Stochastic can be much faster when D is very large

Intermediate approach: use error over subsets of D

Training Linear Regression



Learn Maximum Conditional Likelihood Estimate

$$W_{MCLE} = \arg\min_W \sum_l (y - f(x;W))^2$$

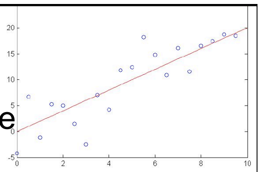
Can we derive gradient descent rule for training?

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

And if $f(x) = w_0 + \sum_i w_i x_i$

$$w_i \leftarrow w_i + \eta \sum_l (y^l - f(x^l;W))\ x_i^l$$

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?

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

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

And if $f(x) = w_0 + \sum_i w_i x_i$ …

Gradient descent rule: $\boxed{w_i \leftarrow w_i + \eta \sum_l (y^l - f(x^l;W))\, x_i^l}$

How about MAP instead of MLE estimate?

Let's assume Gaussian prior: each $w_i \sim N(0, \sigma)$

$$p(w_i) = \frac{1}{Z} \exp\left(-\frac{(w_i - 0)^2}{2\sigma^2} \right)$$

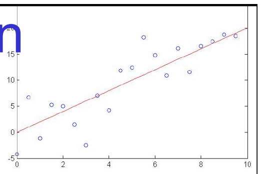
Then MAP estimate is

$$W = \arg\max_{W} -\frac{1}{2\sigma^2} \sum_{w_i \in W} w_i^2 + \sum_{l \in training\ data} \ln P(Y^l | X^l; W)$$

$$= \arg\min_{W} \frac{1}{2\sigma^2} \sum_{w_i \in W} w_i^2 + \sum_{l \in training\ data} (y^l - f(x^l;W))^2$$

Gradient descent: $\boxed{w_i \leftarrow w_i - \lambda w_i + \eta \sum_l (y^l - f(x^l;W))\, x_i^l}$

Consider Linear Regression



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

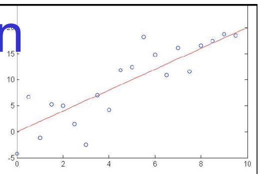
E.g., assume f(x) is linear function of x

$$f(x) = w_0 + \sum_i w_i x_i$$

$$p(y|x) = N(w_0 + \sum_i w_i x_i, \sigma)$$

$$\boxed{w_i \leftarrow w_i - \lambda w_i + \eta \sum_l (y^l - f(x^l; W))\ x_i^l}$$

Consider Linear Regression



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

E.g., assume f(x) is linear function of $\boxed{\phi_i(x)}$

$$f(x) = \sum_i w_i \phi_i(x)$$

$$p(y|x) = N\left(\sum_i w_i \phi_i(x),\ \sigma\right)$$

$$\boxed{w_i \leftarrow w_i - \lambda w_i\ + \eta \sum_l (y^l - f(x^l; W))\ \phi_i(x^l)}$$

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 probability, P(W | data)
4. Again, we can use gradient descent as a general learning algorithm
   - as long as our objective fn is differentiable wrt W
5. Nothing we said here required that f(x) be linear in x -- just linear in W
6. Gradient descent is just one algorithm – linear algebra solutions too

---

Decomposition of Error in Learned Hypothesis

1. Bias

2. Variance

3. Unavoidable error

Bias – Variance decomposition of error

Reading: Bishop chapter 3.2 (different notation)

- Consider simple regression problem f:X→Y

$$y = f(x) + \varepsilon$$

noise N(0,σ)

deterministic

What is expected error of a hypothesis learned (estimated) from randomly drawn training data D?

$$E_D\left[\int_y \int_x (h_D(x) - y)^2 p(y|x)p(x)\ dydx\right]$$

learned estimate of f(x), from training data D

Sources of error

- What if we have perfect learner, infinite data?
  - Our learned h(x) satisfies h(x)=f(x)
  - Still have remaining, *unavoidable error* because of ε

    y = f(x) + ε, $\varepsilon \sim$ N(0,σ)

$$[h(x) = f(x)] \rightarrow [(h(x) - y)^2 = \sigma^2]$$

Sources of error

- What if we have only n training examples?
- What is our expected error
  - Taken over random training sets of size n, drawn from distribution $D=p(x,y)$?

Bias and Variance

given some estimator A for some parameter θ, we define

$$bias(A) = E[A] - \theta$$
$$var(A) = E[(A - E[A])^2]$$

e.g., θ is probability of heads for a coin, A is the MLE estimate for θ, based on *n* independent coin flips

A is a random variable, sampled by reflipping the coins

Expected value is taken over different reflippings

is A biased or unbiased estimator for θ ?

variance decreases as sqrt(1/*n*)

Decomposition of error: $y = f(x) + \varepsilon$; $\varepsilon \sim N(0,\sigma)$

learned estimate of f(x), from training data D

$$E_D\left[\int_y \int_x (h_D(x) - y)^2 p(y|x)p(x)\ dydx\right]$$

$$= unavoidableError + bias^2 + variance$$

$$unavoidableError = \sigma^2$$

$$bias^2 = \int_x (E_D[h_D(x)] - f(x))^2\ p(x)dx$$

$$variance = \int_x E_D[(h_D(x) - E_D[h_D(x)])^2]\ p(x)dx$$

Error Decomposition: Summary

Expected true error of learned P(y|x) for regression (and similarly for classification) has three sources:

1. Unavoidable error
   - non-determinism in world prevents perfect predictions
2. Bias
   - even with infinite training data, hypothesis h(x) might not equal true f(x). E.g., if learner's hypothesis representation cannot represent the true f(x)
3. Variance
   - Whenever we have only finite training data, the sample of just n training examples might represent an empirical distribution that varies from the true P(Y|X). i.e., if we collect many training sets of size n, the empirical distribution they represent will vary about P(Y|X).