THEORY

OF

LINEAR REGRESSION

This page intentionally left blank

credits: T. Mitchell

Regression

- So far, we've been intersted in learning $P(Y|X)$ where Y has discrete values. This is a.k.a. 'classification'.
- What if Y is continious? (a.k.a. 'regression') predict real-valued attributes (e.g. weight, height).
- Predict stock market tomorrow.

Setting

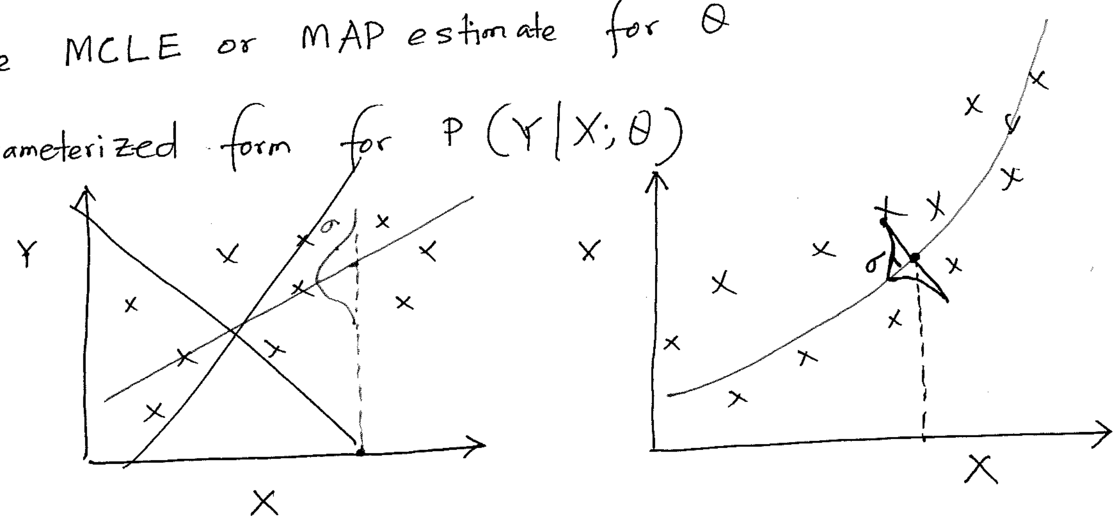
Learn $f: X \rightarrow Y$, where Y is real-valued, given $\{\langle x^1, y^1\rangle, \ldots \langle x^n, y^n\rangle\}$

Approach

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

II. Use MCLE or MAP estimate for θ

I. Choose parameterized form for $P(Y|X;\theta)$



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

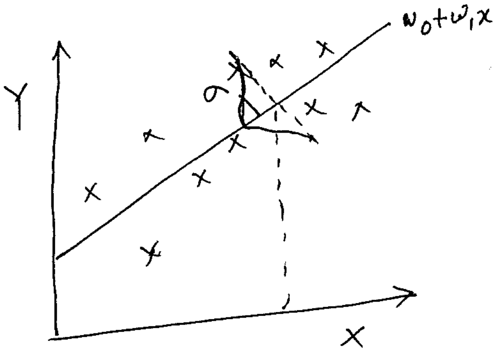
$$y = f(x) + \epsilon \qquad \text{where } \epsilon \sim N(\phi, \sigma)$$

Therefore, Y is a random variable that follows the distribution

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

And expected value of y for any given x is $f(x)$

Linear Regression



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

e.g. assume $f(x)$ is a linear function of x

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

$$\mathbb{E}[y|x] = w_0 + w_1 x$$

$W = \langle w_0, w_1 \rangle$ denotes the parameters (weights)

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

Training Linear Regression

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

Q. How can we learn W from the training data?

A. Learn Maximum Conditional Likelihood Estimate (MCLE)

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

l^{th} training examples

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

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

$$p(y^l | x^l, W) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{y^l - (w_0 + w_1 x^l)}{\sigma}\right)^2}$$

Logistic Regression More Generally

$\vec{X} = \langle x_1, x_2, \dots x_n \rangle$

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

$$f(x) = w_0 + \sum_{i=1}^{n} w_i x_i$$

$$p(y|x) = N\left(w_0 + \sum_{i=1}^{n} w_i x_i, \sigma I\right)$$

$$E[y|x] = w_0 + \sum_{i=1}^{n} w_i x_i$$

$W = \langle w_0, w_1, \dots w_n \rangle$

$$p(y|x; W) = N\left(w_0 + \sum_{i=1}^{n} w_i X_i, \sigma I\right)$$

MCLE derivation

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

where $$p(y^l \mid x^l, W) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{1}{2}\left(\frac{y^l - (w_0 + w_1 x^l)}{\sigma}\right)^2}$$

$$W_{MCLE} = \arg\max_{W} \sum_{l} \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2}\left(\frac{y^l - (w_0 + w_1 x^l)}{\sigma}\right)^2$$

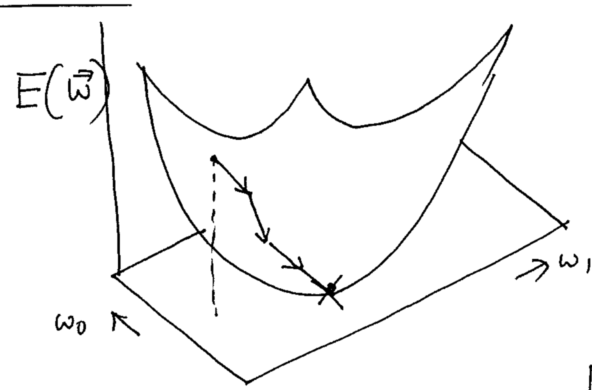
$$W_{MCLE} = \arg\max_{W} \sum_{l} -\frac{1}{2\sigma^2}\left(y^l - (w_0 + w_1 x^l)\right)^2$$

$$W_{MCLE} = \arg\underline{\min}_{W} \sum_{l} \boxed{\frac{1}{2\sigma^2}}\left(y^l - (w_0 + w_1 x^l)\right)^2$$

$$= \arg\min_{W} \sum_{l} \underbrace{\left(y^l - (w_0 + w_1 x^l)\right)^2}_{\text{"squared error"}}$$

[often used in "curve fitting"]

Gradient Descent :



Gradient $$\nabla E(\vec{w}) = \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \ldots \frac{\partial E}{\partial w_n}\right]$$

Training Rule: $$\vec{w}^{(i+1)} \leftarrow \vec{w}^{(i)} \underbrace{-}_{\text{opposite of gradient}} \underbrace{\eta}_{\text{learning rate}} \nabla E(\vec{w})$$

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

Minimizing Squared Error: Gradient Descent

$$\frac{\partial E}{\partial w_1} = \sum_{l} 2\left(y^l - (w_0 + w_1 x^l)\right)(-x^l)$$

$$= -2\sum_{l}\left(y^l - (w_0 + w_1 x^l)\right) x^l$$

$$\frac{\partial E}{\partial w_0} = -2\sum_{l}\left(y^l - (w_0 + w_1 x^l)\right)$$

So, our update rule is:

$$w_0 \leftarrow w_0 - \eta\left[-2\sum_{l}\left(y^l - (w_0 + w_1 x^l)\right)\right]$$

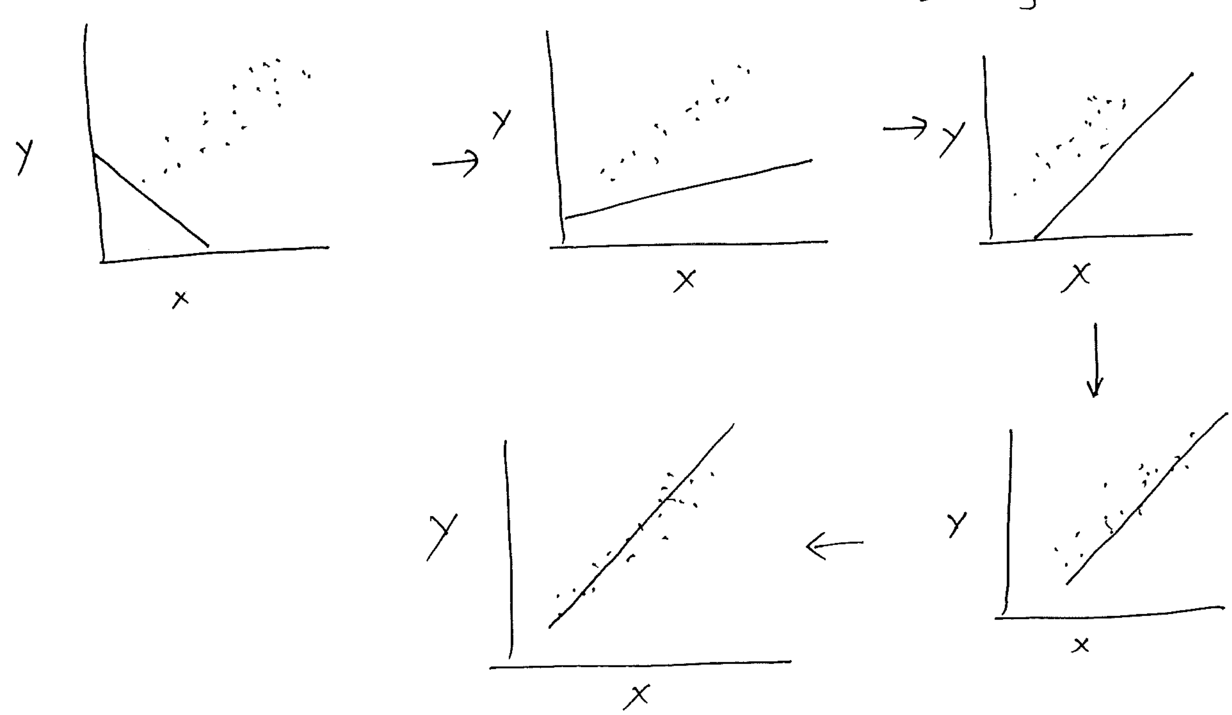
$$w_0 \leftarrow w_0 + 2\eta\left[\sum_{l}\left(y^l - (w_0 + w_1 x^l)\right)\right]$$

$$w_1 \leftarrow w_1 - \eta\left[-2\sum_{l} x^l\left(y^l - (w_0 + w_1 x^l)\right)\right]$$

$$w_1 \leftarrow w_1 + 2\eta\left[\sum_{l} x^l\left(y^l - (w_0 + w_1 x^l)\right)\right]$$

In general,

$$w_i \leftarrow w_i + 2\eta \sum_{l} x_i^l \Big(\underbrace{y^l - \sum_{j=1}^{n} w_j x_j^l}_{\text{"misassignment"}}\Big)$$



"converged

Gradient Descent More Generally

Iterate until change $< \epsilon$

$\forall i$ repeat

$$w_i \leftarrow w_i + \eta 2 \sum_l x_i^l \left(Y^l - \left(w_0 + \sum_{j=1}^{n} w_j x_j^l \right) \right)$$

assume $x_\phi = 1$ for w_0

How about MAP estimate insted of MCLE ?

$$W_{MAP} = \arg\max_W \ln N(W \mid \phi, I) + \sum_l \ln \left(P(Y^l \mid X^l, W) \right)$$

$$= \arg\max_W \underbrace{\left(-c \sum_l w_i^2 \right)}_{\substack{\text{regulization term} \\ \text{pushes } W \text{ to zero}}} + \sum_l \ln \left(P(Y^l \mid X^l, W) \right)$$

Bias and Variance

Given algorithm that outputs estimate $\hat{\theta}$ for θ,

the bias of the estimator : $E[\hat{\theta}] - \theta$

the variance of the estimator : $E[(\hat{\theta} - E(\hat{\theta}))^2]$

e.g. $\hat{\theta}$ MLE estimator for probability θ of heads, based on n independent coin flips

$$\hat{\theta}^{MLE} = \frac{\alpha_1}{\alpha_1 + \alpha_0}$$

What is the bias? $\varnothing = E[\hat{\theta}^{MLE}]$

What is the variance? $\frac{\theta(1-\theta)}{n} = Var[\hat{\theta}^{MLE}] = E[\underbrace{(\hat{\theta}^{MLE} - \theta)^2}_{\text{squared error}}]$

Let's use $\hat{\theta}$ MAP estimator

$$\hat{\theta}^{MAP} = \frac{\alpha_1 + \beta_1 - 1}{(\alpha_1 + \beta_1 - 1) + (\alpha_0 + \beta_0 - 1)}$$

Which estimator has higher bias? higher variance? (for finite n)

Bias

$\hat{\theta}^{MAP}$ has higher bias than $\hat{\theta}^{MLE}$

Variance

$\hat{\theta}^{MLE}$ has higher variance.

since the fraction will vary more than the fraction in $\hat{\theta}^{MAP}$

As $n \to \infty$, $\hat{\theta}^{MAP} \to \hat{\theta}^{MLE}$

so, bias approaches zero

so, variance, approaches zero

Train Set error :	1%	15%	15%	0.5%
CV Set error :	11%	16%	30%	1%
	high variance (overfitting)	high bias (underfitting)	(high bias high variance	low bias low variance