Machine Learning

CSE 6363 (Fall 2019)

Lecture 9 Logistic Regression

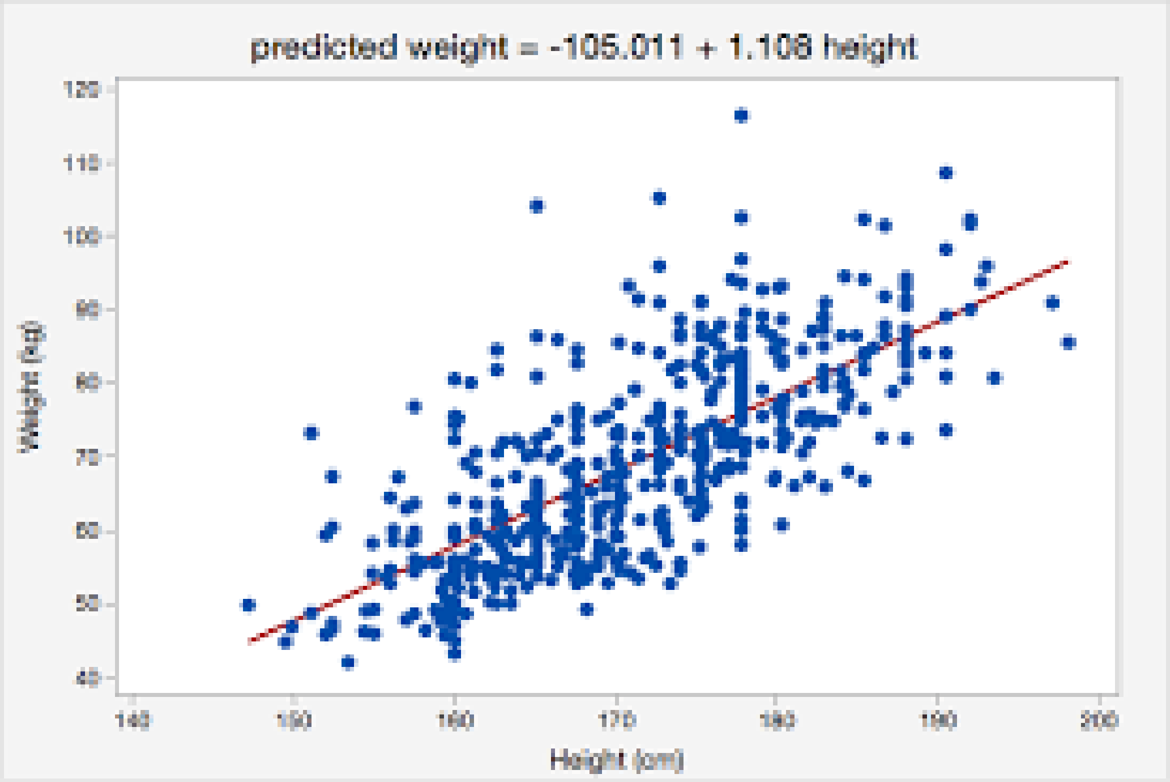
Dajiang Zhu, Ph.D.

Department of Computer Science and Engineering

Slides of this course (CSE6363) courtesy: Dr. Heng Huang, Dr. Aarti Singh

Why Logistic Regression

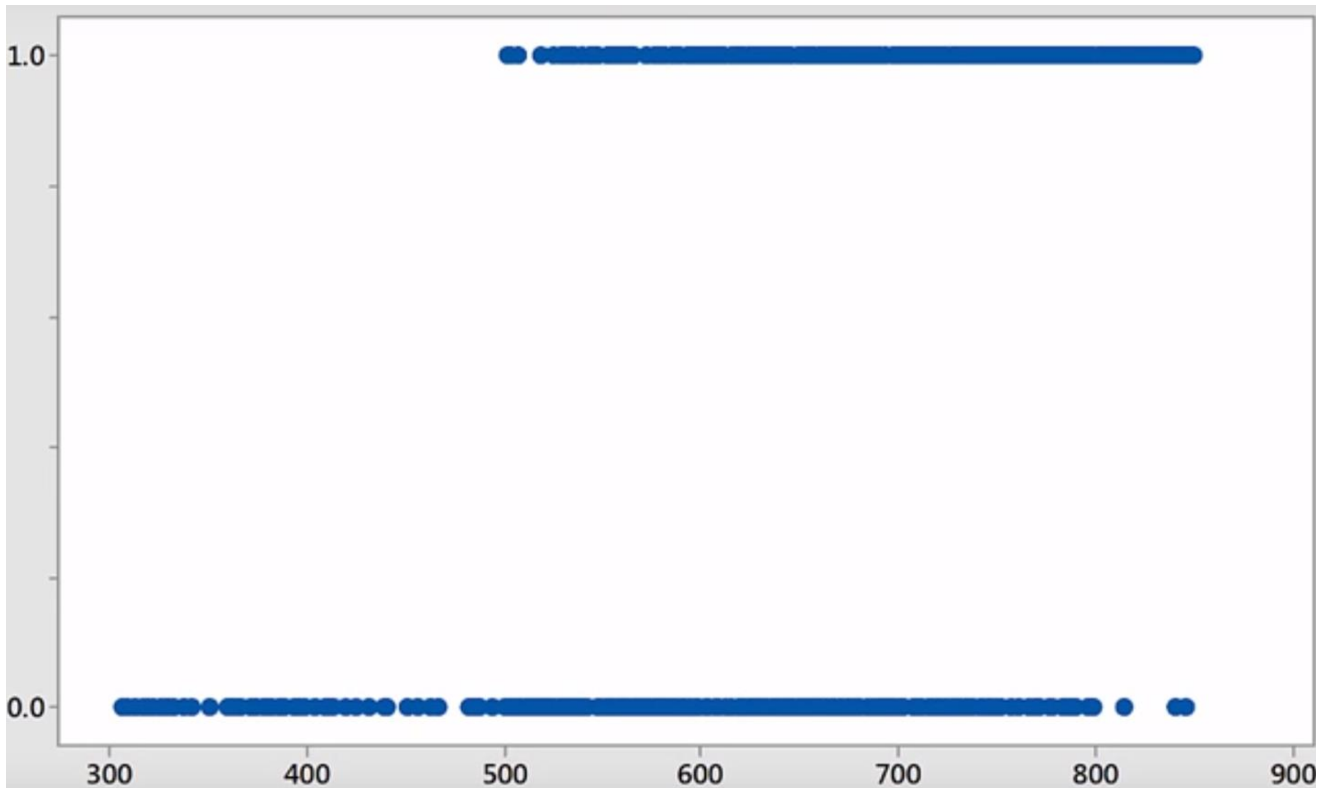
- Ordinary Regression



Why Logistic Regression

Get offer ☺

Go home ☹

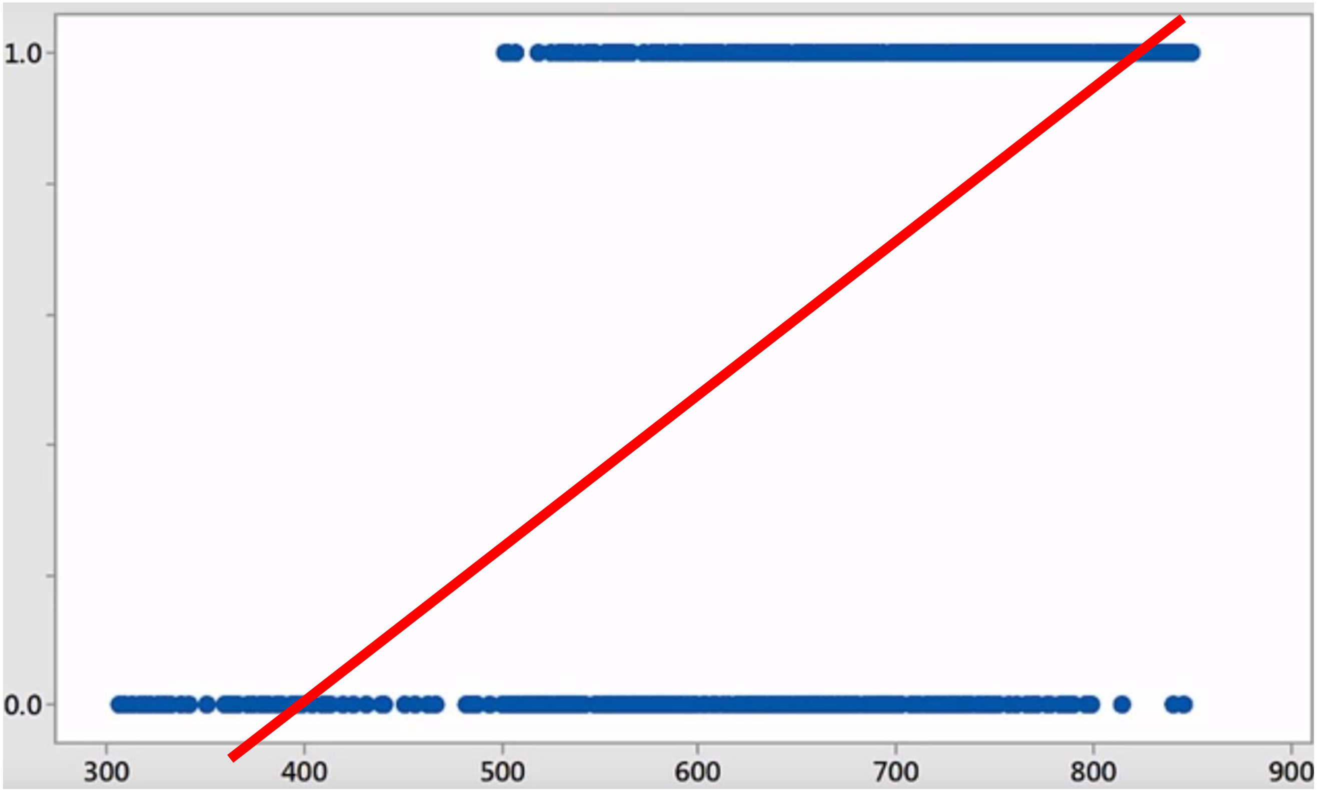


Review scores

Why Logistic Regression

Get offer ☺

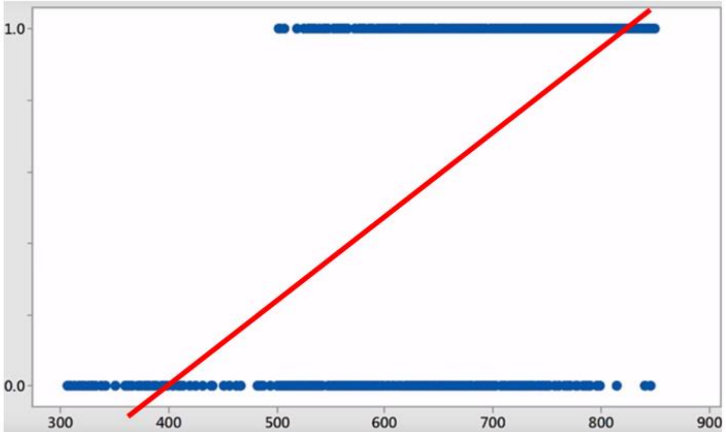
Go home ☹



Review scores

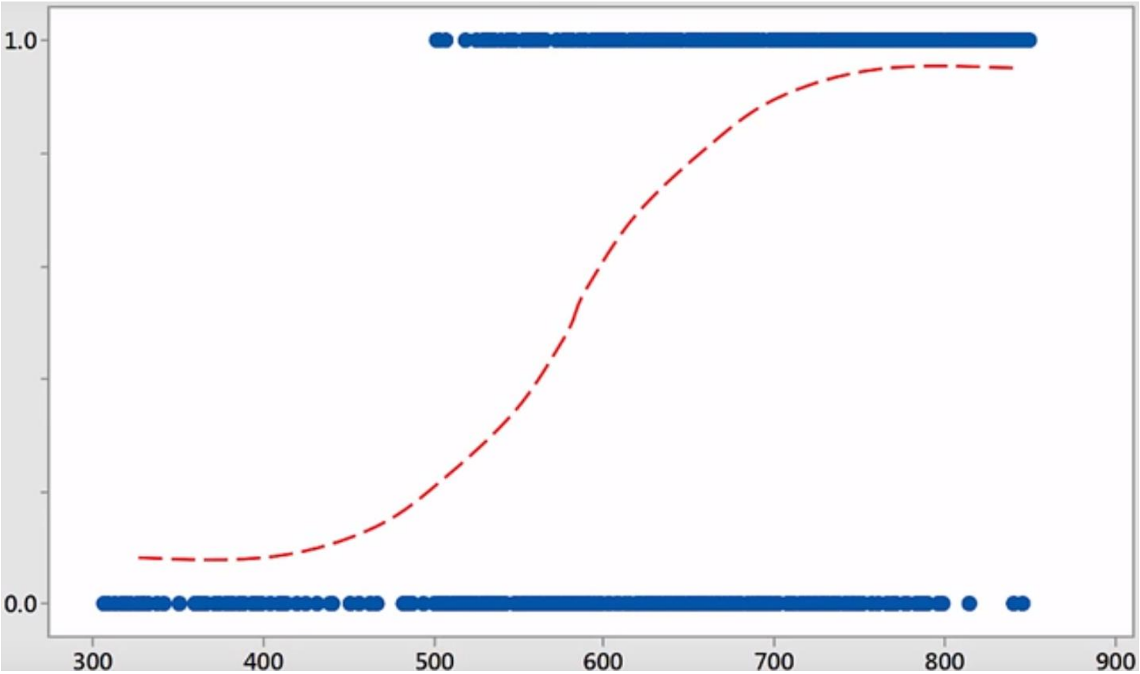
Sum squared error $\sum_i (X_i^\top w - y_i)^2$

Why Logistic Regression



- Error are NOT normally distributed (error has pattern!!!)
- Predicted Y should be 0 or 1. It should be better than Mean!

Why Logistic Regression



- Error are NOT normally distributed (error has pattern!!!) - **Solved**
- Predicted Y should be 0 or 1. It should be better than Mean! – **Solved**
- **Moreover, we are predicting Probability!**

Probability and Odds

$$P = \frac{Outcomes\ of\ Interest}{All\ Possible\ Outcomes}$$

Fair coin flip: P (heads)?
Fair die roll: P (1 or 2)?
Deck of playing cards: P (diamond card)?

Probability and Odds

$$odds = \frac{P\ (occurring)}{P\ (not\ occurring)}$$

Fair coin flip: odds (heads): 1

Fair die roll: P (1 or 2): 0.5

Deck of playing cards: P (diamond card): 1/3

What is the logit

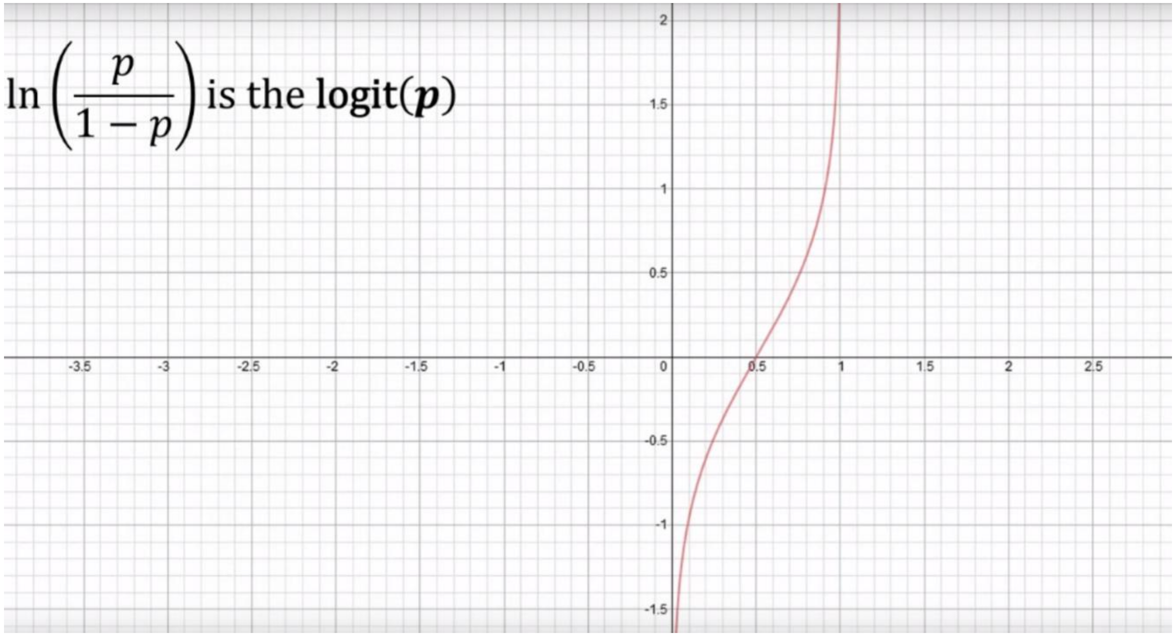
- The goal of logistic regression is to estimate *p* for a linear combination of the independent variables.

- To tie together our linear combination of variables and in essence Binomial distribution, we need a function to link them together. That means we need a mapping function to map the linear combination of variables that could result in any value onto the Binomial probability distribution with a domain from 0 to 1.

- The natural log of the odds ratio – the logit- is the link function.

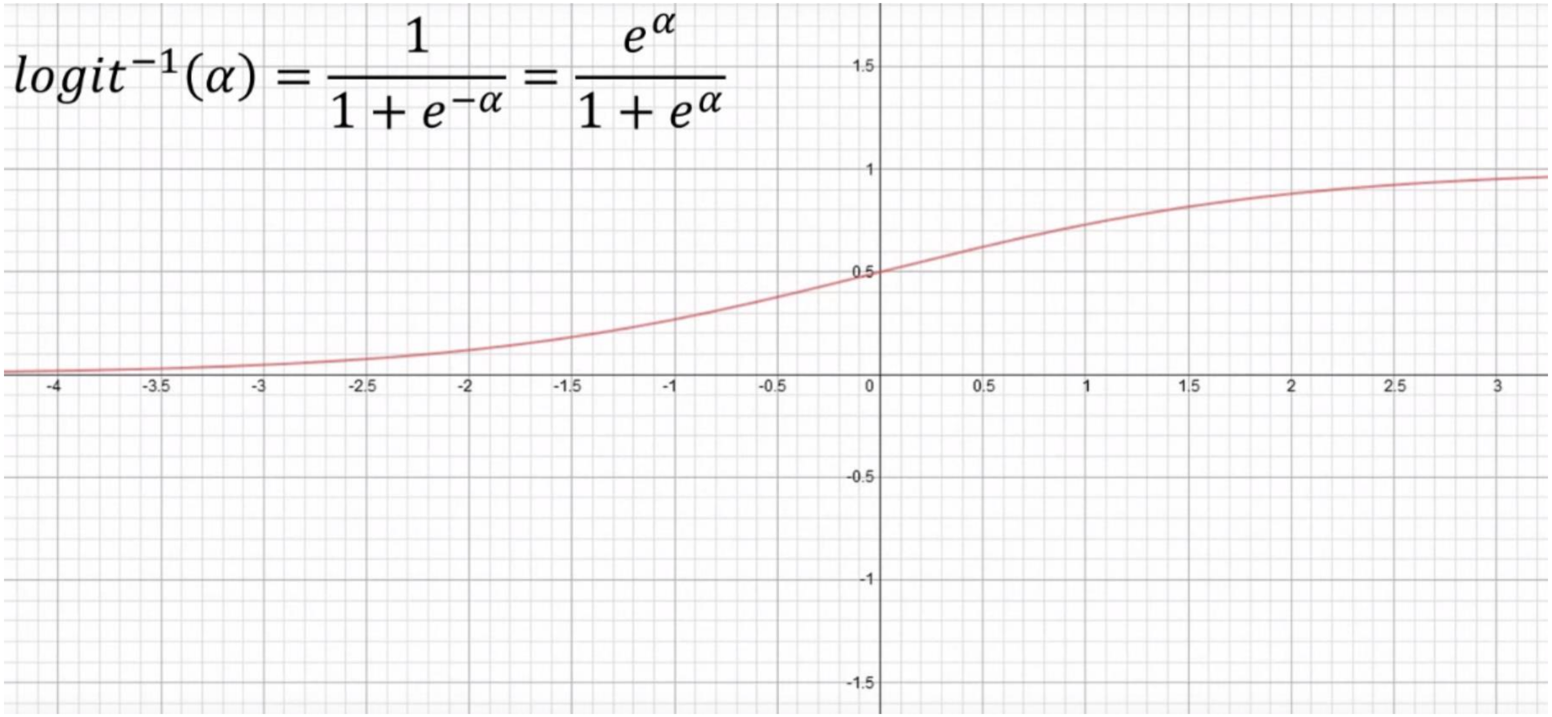
What is the logit

$$logit(p) = \ln(\frac{p}{1-p})$$



Why Logistic Regression

$$logit(p) = \ln\left(\frac{p}{1-p}\right) = \beta x$$



Logistic Regression

Basic idea:

Regression $\rightarrow$ Calculate p

Generative vs. Discriminative Classifiers

Generative classifiers (e.g. Naïve Bayes)

•Assume some functional form for P(X,Y) (or P(X|Y) and P(Y))

•Estimate parameters of P(X|Y), P(Y) directly from training data

•Use Bayesrule to calculate P(Y|X)

Why not learn P(Y|X) directly? Or better yet, why not learn the decision boundary directly?

Discriminative classifiers (e.g. Logistic Regression)

•Assume some functional form for P(Y|X) or for the decision boundary

•Estimate parameters of P(Y|X) directly from training data

Logistic Regression

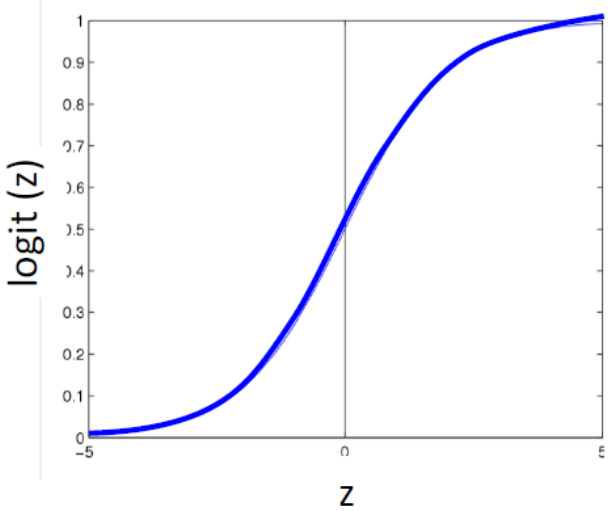
Assumes the following functional form for P(Y|X):

$$P(Y=1|X) = \frac{1}{1+\exp(w_0 + \sum_i w_i X_i)}$$

Logistic function applied to a linear function of the data

Logistic function (or Sigmoid):

$$\frac{1}{1+exp(-z)}$$

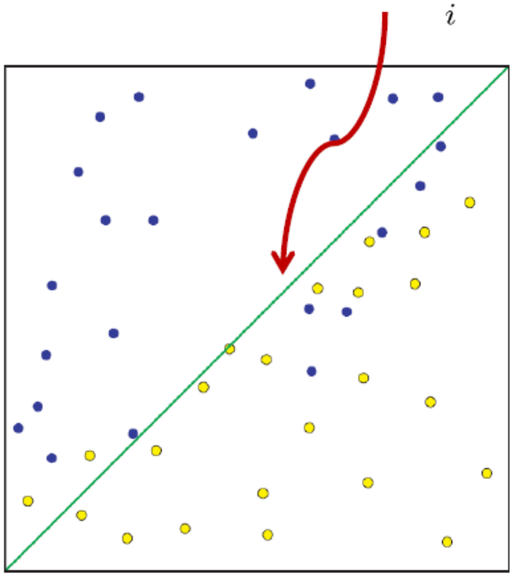


Logistic Regression is a Linear Classifier

Assumes the following functional form for P(Y|X):

$$P(Y=1|X) = \frac{1}{1+\exp(w_0+\sum_i w_i X_i)}$$

$$w_0 + \sum_i w_i X_i = 0$$



$$w_0 + \sum_i w_i X_i \overset{0}{\underset{1}{\gtrless}} 0$$

$$P(Y=0|X) \overset{0}{\underset{1}{\gtrless}} P(Y=1|X)$$

Logistic Regression

Assumes the following functional form for P(Y|X):

$$P(Y=1|X) = \frac{1}{1+\exp(w_0+\sum_i w_i X_i)}$$

$$\Rightarrow P(Y=0|X) = \frac{\exp(w_0+\sum_i w_i X_i)}{1+\exp(w_0+\sum_i w_i X_i)}$$

$$\Rightarrow \frac{P(Y=0|X)}{P(Y=1|X)} = \exp(w_0+\sum_i w_i X_i) \quad \overset{0}{\underset{1}{\gtrless}} \quad 1$$

$$\Rightarrow \boxed{w_0+\sum_i w_i X_i \quad \overset{0}{\underset{1}{\gtrless}} \quad 0}$$

Logistic Regression

We'll focus on binary classification
– 0 or 1 for Y

$$P(Y=0|\mathbf{X},\mathrm{w}) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

$$P(Y=1|\mathbf{X},\mathrm{w}) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

Goal: Learn the parameters w0, w1, … wd **directly**!
-------------------But how?

Two Strategies

- Maximum Likelihood Estimation (MLE)
  - Maximizes the probability of observed data

- Maximum A Posteriori Estimation (MAP)
  - Maximizes a posterior probability

Ref: Carlos Guestrin

Maximum Likelihood Estimation

- **Data:** Observed set D of α_H Heads and α_T Tails
- **Hypothesis:** Binomial distribution
- Learning θ is an optimization problem
  - What's the objective function?
- MLE: Choose θ that maximizes the probability of observed data:

$$\begin{aligned}\hat{\theta} &= \arg\max_{\theta} \; P(\mathcal{D} \mid \theta) \\ &= \arg\max_{\theta} \; \ln P(\mathcal{D} \mid \theta)\end{aligned}$$

Ref: Carlos Guestrin

Logistic Regression

Training Data $\{(X^{(j)}, Y^{(j)})\}_{j=1}^{n}$ $X^{(j)} = (X_1^{(j)}, \ldots, X_d^{(j)})$

Maximum (**Conditional**) Likelihood Estimates

$$\widehat{\mathbf{w}}_{MCLE} = \arg\max_{\mathbf{w}} \prod_{j=1}^{n} P(Y^{(j)} \mid X^{(j)}, \mathbf{w})$$

Discriminative method – Don’t waste effort learning P(X), focus on P(Y|X) –that’s all that matters for classification!

Logistic Regression

$$P(Y=0|\mathbf{X},\mathbf{w}) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

$$P(Y=1|\mathbf{X},\mathbf{w}) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

$$\begin{aligned} l(\mathbf{w}) &\equiv \ln \prod_j P(y^j|\mathbf{x}^j,\mathbf{w}) \\ &= \sum_j \left[y^j (w_0 + \sum_i^d w_i x_i^j) - \ln(1+exp(w_0+\sum_i^d w_i x_i^j)) \right] \end{aligned}$$

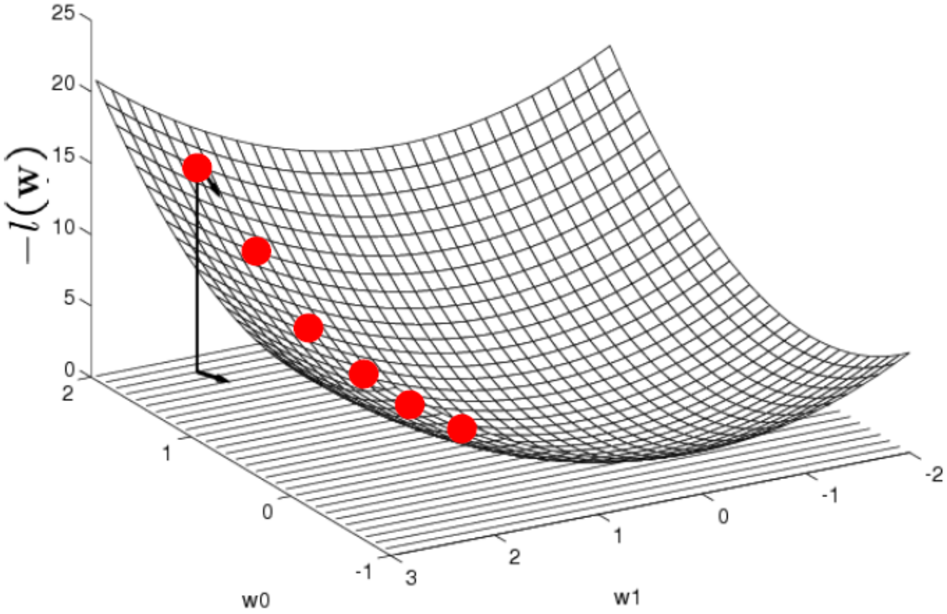
No closed-form solution to maximize *l*(**w**)

Logistic Regression

- Conditional likelihood for Logistic Regression is concave
- Maximum of a concave function = minimum of a convex function

Gradient Ascent (concave)/ Gradient Descent (convex)

Logistic Regression



Gradient:

$$\nabla_{\mathbf{w}} l(\mathbf{w}) = [\frac{\partial l(\mathbf{w})}{\partial w_0}, \ldots, \frac{\partial l(\mathbf{w})}{\partial w_n}]'$$

Update rule:

Learning rate, η>0

$$\Delta \mathbf{w} = \eta \nabla_{\mathbf{w}} l(\mathbf{w})$$

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \frac{\partial l(\mathbf{w})}{\partial w_i}\Big|_t$$

Gradient Ascent for Logistic Regression

Gradient ascent algorithm:

iterate until change < ε

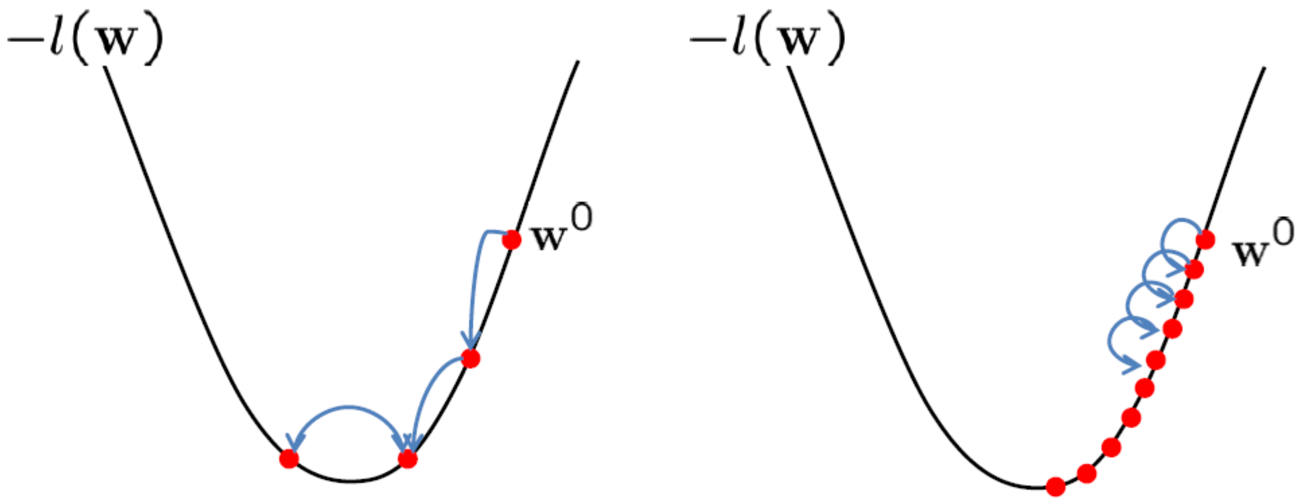
$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \sum_j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})]$$

For i=1,…,d,

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - \boxed{\hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})}]$$

Predict what current weight thinks label Y should be

Effect of step-size



Large η => Fast convergence but larger residual error
Also possible oscillations

Small η => Slow convergence but small residual error

Maximum A Posteriori (MAP) Estimation

- MAP estimation picks the mode of the posterior

$$\hat{\theta}_{MAP} = \arg\max_{\theta} p(D|\theta)p(\theta)$$

- If $\theta \sim Be(a, b)$, this is just

$$\hat{\theta}_{MAP} = (a-1)/(a+b-2)$$

- MAP is equivalent to maximizing the penalized maximum log-likelihood

$$\hat{\theta}_{MAP} = \arg\max_{\theta} \log p(D|\theta) - \lambda c(\theta)$$

where $c(\theta) = -\log p(\theta)$ is called a *regularization term*. λ is related to the strength of the prior.

How about MAP?

$$p(\mathbf{w} \mid Y, \mathbf{X}) \;\propto\; P(Y \mid \mathbf{X}, \mathbf{w})p(\mathbf{w})$$

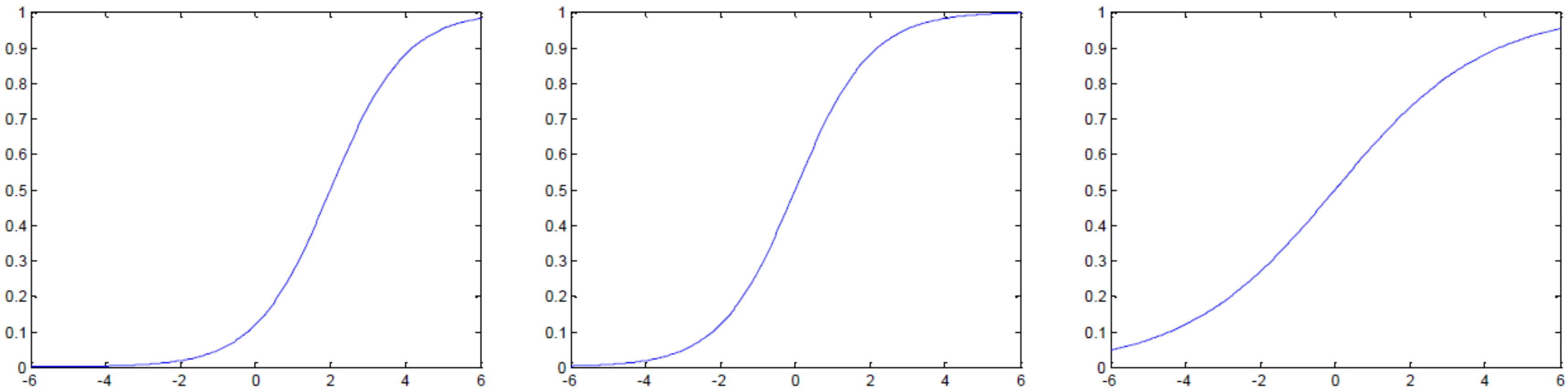
- One common approach is to define priors on **w**
  - ✓ Normal distribution, zero mean, identity covariance
  - ✓ "Pushes" parameters towards zero
- Corresponds to ***Regularization***
  - ✓ **Helps avoid very large weights and overfitting**
- M(C)AP estimate:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \ln \left[p(\mathbf{w}) \prod_{j=1}^{n} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

Understanding the sigmoid

$$g(w_0 + \sum_i w_i x_i) = \frac{1}{1 + e^{w_0 + \sum_i w_i x_i}}$$

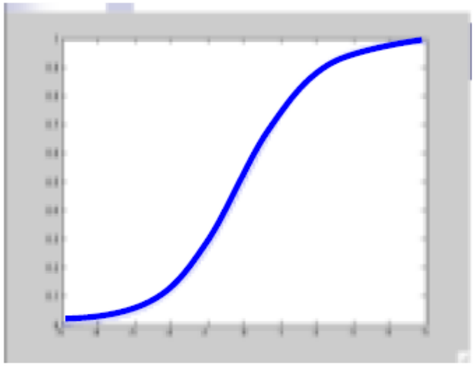
w_0=-2, w_1=-1 $\quad$ w_0=0, w_1=-1 $\quad$ w_0=0, w_1=-0.5



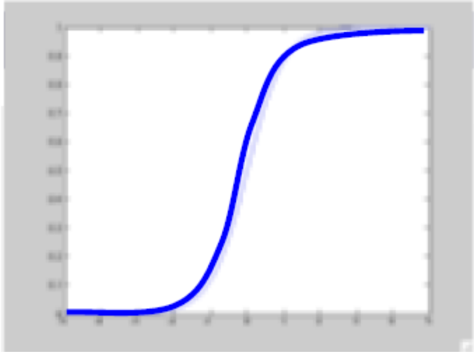
$$z = w_0 + \sum_i w_i x_i$$

Understanding the sigmoid

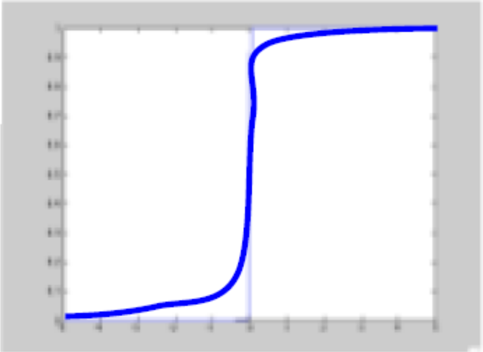
Large weights ➡ Overfitting



$$\frac{1}{1+e^{-x}}$$



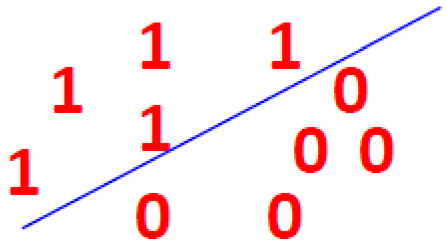
$$\frac{1}{1+e^{-2x}}$$



$$\frac{1}{1+e^{-100x}}$$

Understanding the sigmoid

- Large weights lead to overfitting:



- Penalizing high weights can prevent overfitting

M(C)AP –Regularization

$$\arg\max_{\mathbf{w}} \ln \left[p(\mathbf{w}) \prod_{j=1}^{n} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

$$p(\mathbf{w}) = \prod_i \frac{1}{\kappa\sqrt{2\pi}} \; e^{\frac{-w_i^2}{2\kappa^2}}$$

Zero-mean Gaussian prior

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \sum_{j=1}^{n} \ln P(y^j \mid \mathbf{x}^j, \mathbf{w}) - \boxed{\sum_{i=1}^{d} \frac{w_i^2}{2\kappa^2}}$$

Will penalizes large weights

M(C)AP –Regularization

Calculate gradient

$$\frac{\partial}{\partial w_i} \ln \left[p(\mathbf{w}) \prod_{j=1}^{n} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

$$\underbrace{\frac{\partial}{\partial w_i} \ln p(\mathbf{w})}_{\propto \frac{-w_i}{\kappa^2}} + \underbrace{\frac{\partial}{\partial w_i} \ln \left[\prod_{j=1}^{n} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]}_{\text{Same as MCLE}}$$

Same as MCLE

$$\propto \frac{-w_i}{\kappa^2}$$

Extra term penalizes large weights

M(C)LE vs. M(C)AP

- Maximum conditional likelihood estimate

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \ln \left[\prod_{j=1}^{n} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

$$\boxed{w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - P(Y=1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})]}$$

- Maximum conditional a posteriori estimate

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \ln \left[p(\mathbf{w}) \prod_{j=1}^{n} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

$$\boxed{w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \left\{ -\frac{1}{\kappa^2} w_i^{(t)} + \sum_j x_i^j [y^j - P(Y=1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})] \right\}}$$

Generative vs Discriminative

Given **infinite data** (asymptotically),

- If conditional independence assumption holds, Discriminative and generative NB perform similar

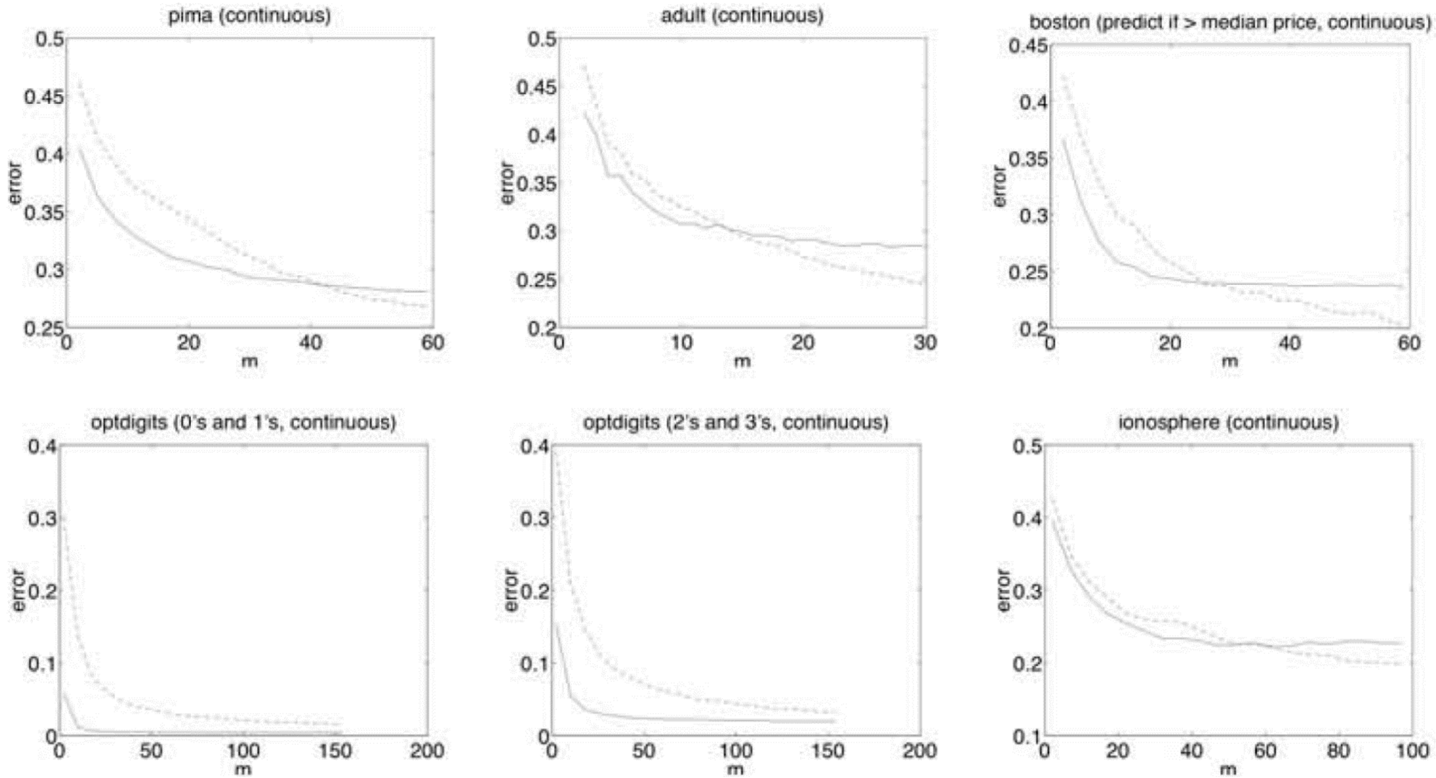
$$\epsilon_{\mathrm{Dis},\infty} \sim \epsilon_{\mathrm{Gen},\infty}$$

- If conditional independence assumption does NOT holds, Discriminative outperforms generative NB

$$\epsilon_{\mathrm{Dis},\infty} < \epsilon_{\mathrm{Gen},\infty}$$

[Ng & Jordan, NIPS 2001]

Naïve Bayes vs Logistic Regression



—— Naïve Bayes ----- Logistic Regression

[Ng & Jordan, NIPS 2001]

Summary

- LR is a linear classifier
  - ➢ decision rule is a hyperplane
- LR optimized by conditional likelihood
  - ➢ no closed-form solution
  - ➢ concave !global optimum with gradient ascent
  - ➢ Maximum conditional a posteriori corresponds to regularization
- In general, NB and LR make different assumptions
  - ➢ NB: Features independent
  - ➢ LR: Functional form of P(Y|X), no assumption on P(X|Y)
- Convergence rates
  - ➢ GNB (usually) needs less data
  - ➢ LR (usually) gets to better solutions in the limit