

# Logistic Regression

Il-Chul Moon  
Dept. of Industrial and Systems Engineering  
KAIST

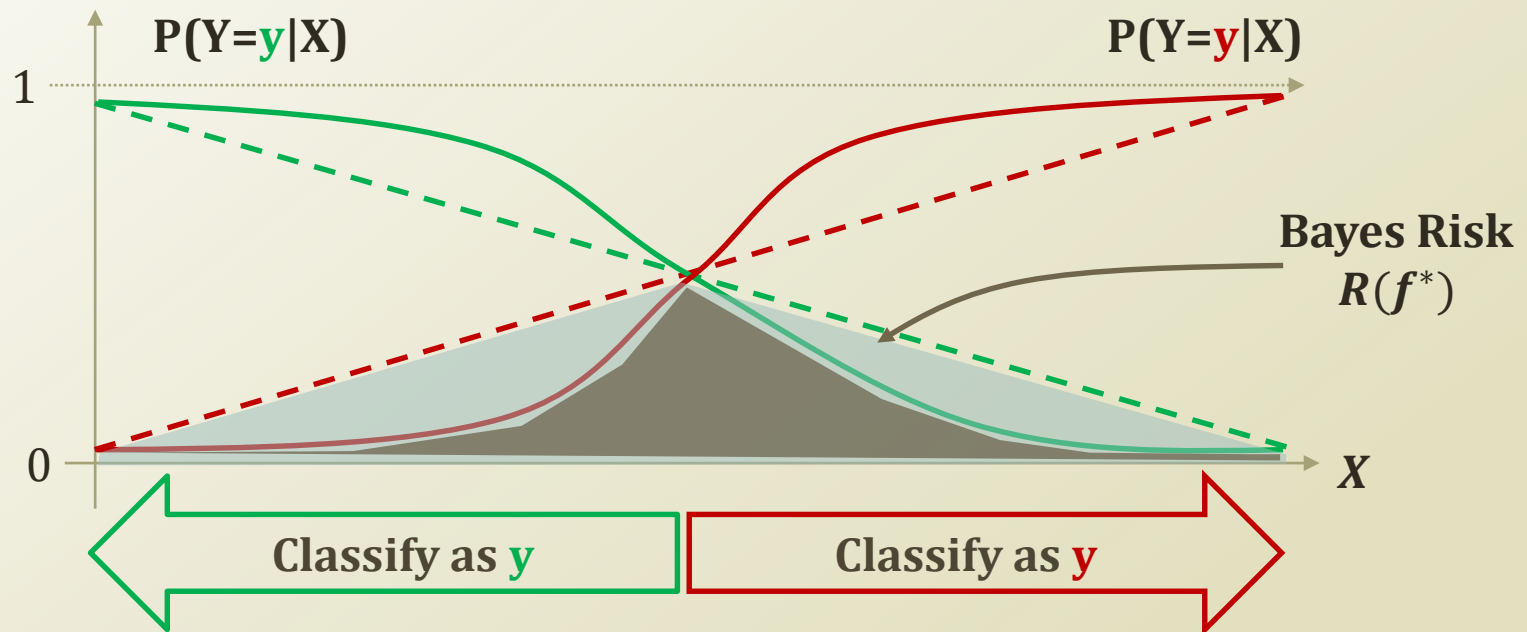
[icmoon@kaist.ac.kr](mailto:icmoon@kaist.ac.kr)

# Weekly Objectives

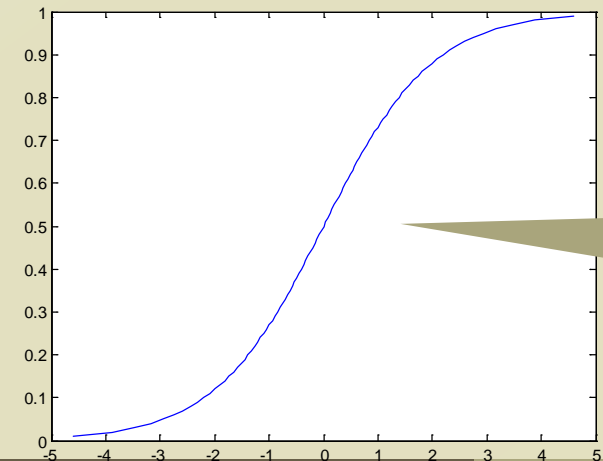
- Learn the logistic regression classifier
  - Understand why the logistic regression is better suited than the linear regression for classification tasks
  - Understand the logistic function
  - Understand the logistic regression classifier
  - Understand the approximation approach for the open form solutions
- Learn the gradient descent algorithm
  - Know the Taylor expansion
  - Understand the gradient descent/ascent algorithm
- Learn the difference between the naïve Bayes and the logistic regression
  - Understand the similarity of the two classifiers
  - Understand the differences of the two classifiers
  - Understand the performance differences

# LOGISTIC REGRESSION

# Optimal Classification and Bayes Risk



- Linear function vs. Non-linear function of  $P(Y|X)$ 
  - Which is better?
- Problems of linear function
  - Range
  - Risk optimization
- Which function to use?
  - Need S-curve!

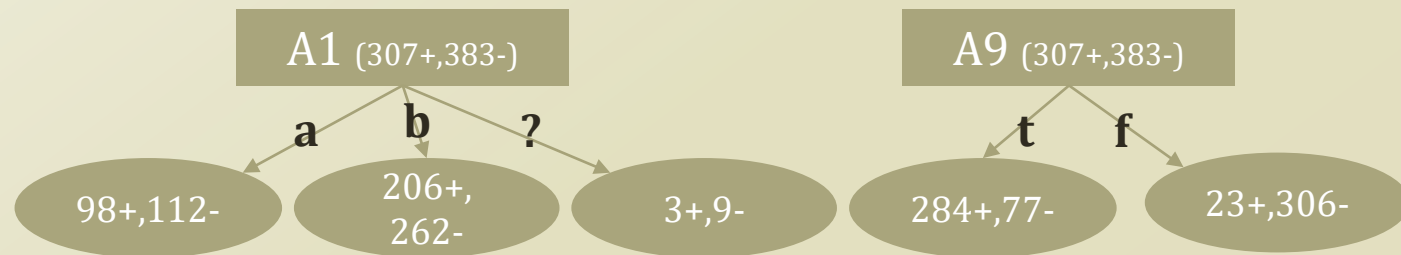


# Detour: Credit Approval Dataset

- <http://archive.ics.uci.edu/ml/datasets/Credit+Approval>
- To protect the confidential information, the dataset is anonymized
  - Feature names and values, as well
- A1: b, a.
- A2: continuous.
- A3: continuous.
- A4: u, y, l, t.
- A5: g, p, gg.
- A6: c, d, cc, i, j, k, m, r, q, w, x, e, aa, ff.
- A7: v, h, bb, j, n, z, dd, ff, o.
- A8: continuous.
- A9: t, f.
- A10: t, f.
- A11: continuous.
- A12: t, f.
- A13: g, p, s.
- A14: continuous.
- A15: continuous.
- **C: +, - (class attribute)**

## Some Counting Result

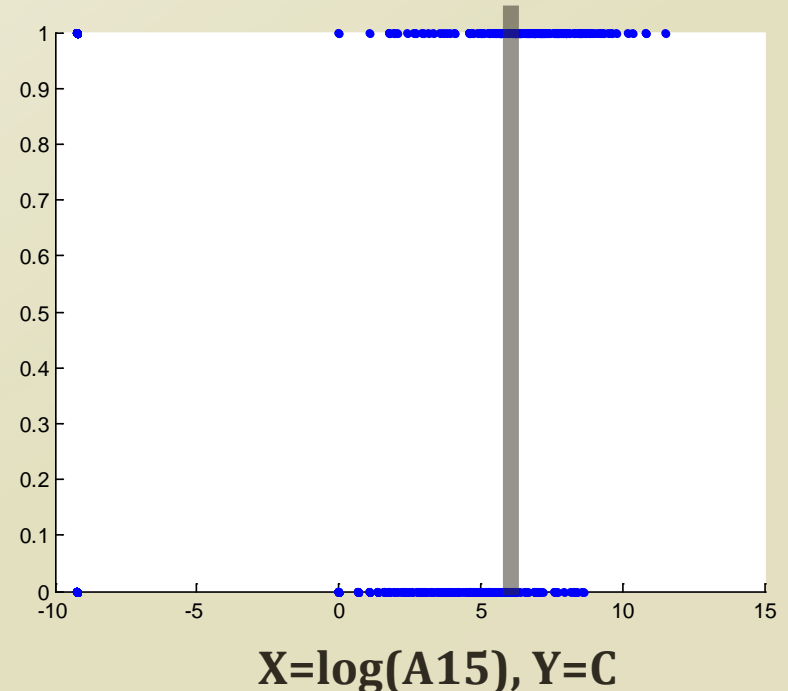
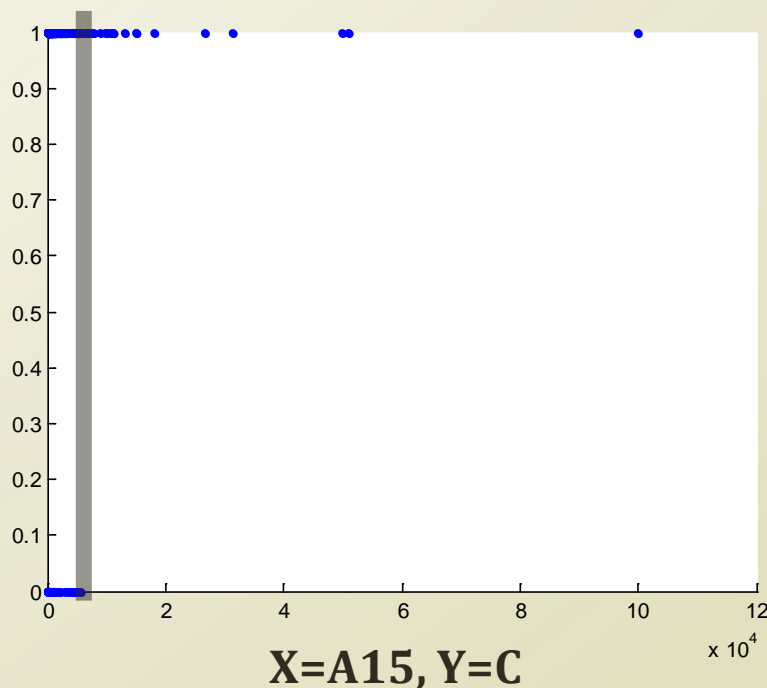
- 690 instances total
- 307 positive instances
- Considering A1
  - 98 positive when a
  - 112 negative when a
  - 206 positive when b
  - 262 negative when b
  - 3 positive when ?
  - 9 negative when ?
- Considering A9
  - 284 positive when t
  - 77 negative when t
  - 23 positive when f
  - 306 negative when f



Which is a better attribute to include in the feature set of the hypothesis?

# Classification with One Variable

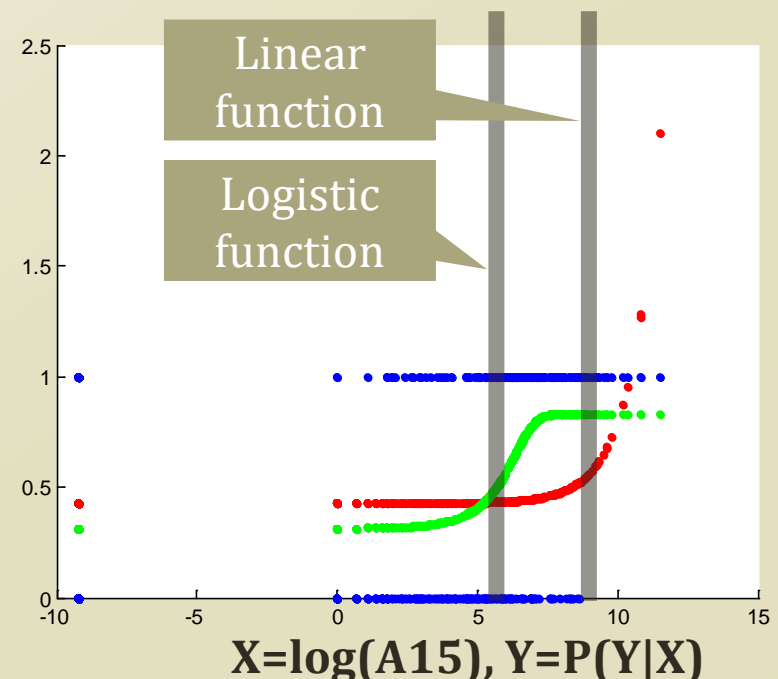
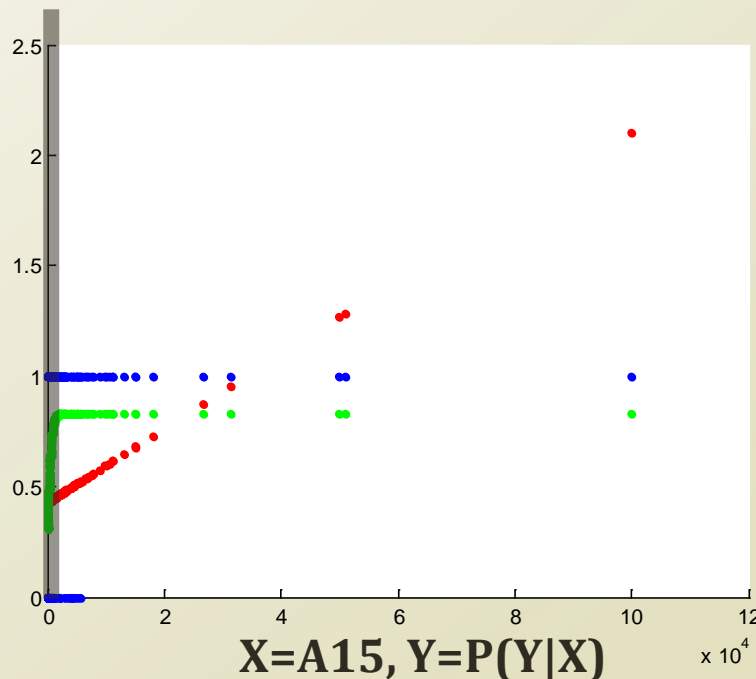
- Let's predict the class,  $C$ , with an attribute,  $A15$ 
  - Imagine that the Y axis shows  $P(Y|X)$
  - There is a decision boundary
    - You can see it intuitively
- Then, How to find the boundary?



# Linear Function vs. Non-Linear Function

- Problem of fitting to the linear function
  - Violate the probability axiom
  - Slow response to the examples
- Better to fit to the logistic function
  - Keep the probability axiom
  - Quick response around the decision boundary
- Which function to use?

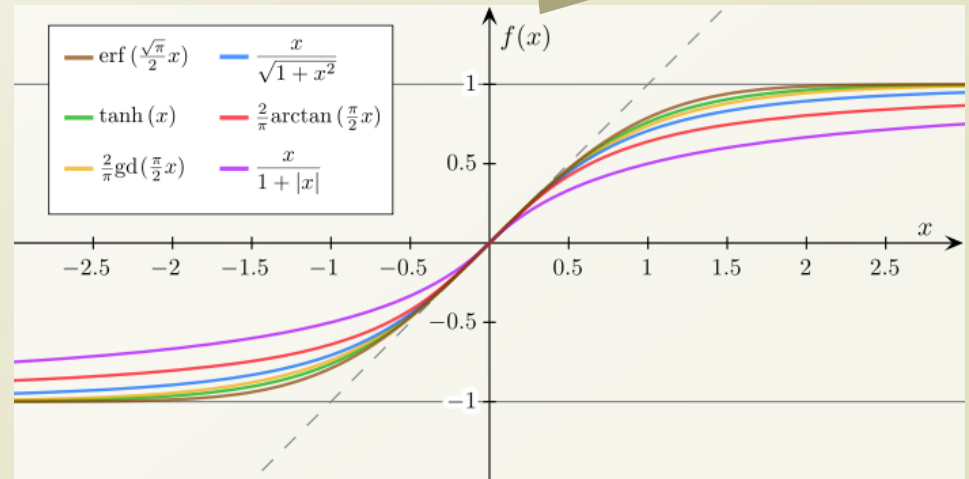
Blue =  $(X, Y_{\text{true}})$ , Red =  $(X, P_{\text{lin}}(Y|X))$ , Green =  $(X, P_{\text{log}}(Y|X))$



# Logistic function

Many types of sigmoid functions

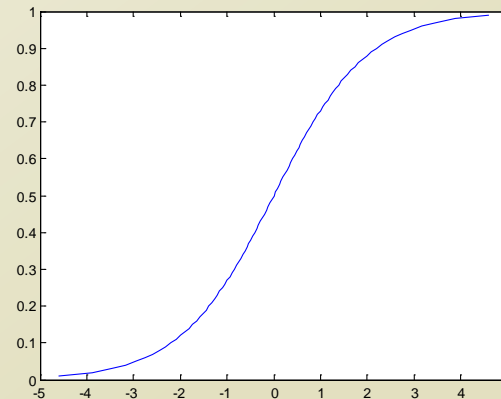
- Sigmoid function is
  - Bounded
  - Differentiable
  - Real function
  - Defined for all real inputs
  - With positive derivative



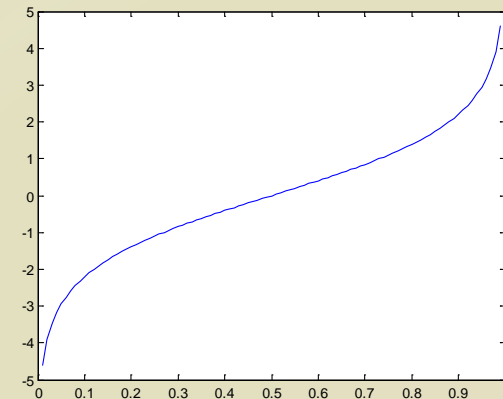
- Logistic function is

- $f(x) = \frac{1}{1+e^{-x}}$
- In relation to the population growth
- Why is this good?

- Sigmoid function
- Particularly, easy to calculate the derivative...



Logistic Function

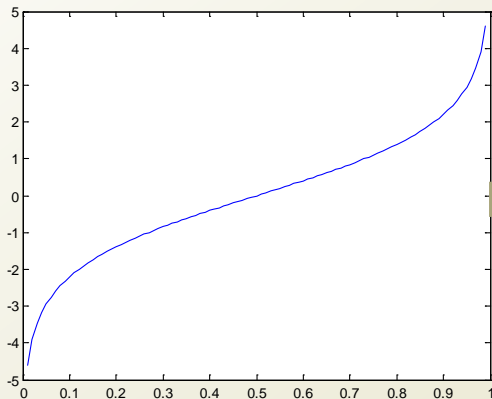


Logit Function

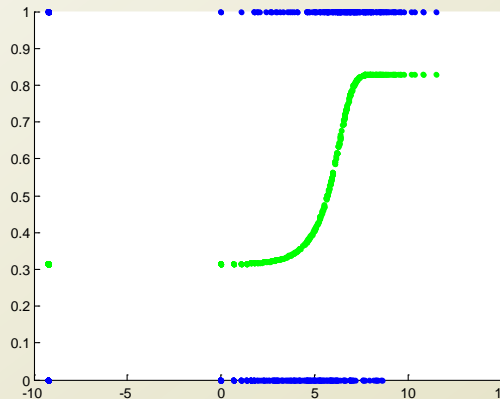
$$f(x) = \log\left(\frac{x}{1-x}\right)$$



# Logistic Function Fitting



**Logit Function**



**Logistic Fitting**

**Linear Regression:**

$$\hat{f} = X\theta \quad \theta = (X^T X)^{-1} X^T Y$$

Very similar to the linear regression.  
Turning to the multivariate case

$$f(x) = \log\left(\frac{x}{1-x}\right) \rightarrow x = \log\left(\frac{p}{1-p}\right) \rightarrow ax + b = \log\left(\frac{p}{1-p}\right) \rightarrow X\theta = \log\left(\frac{p}{1-p}\right)$$

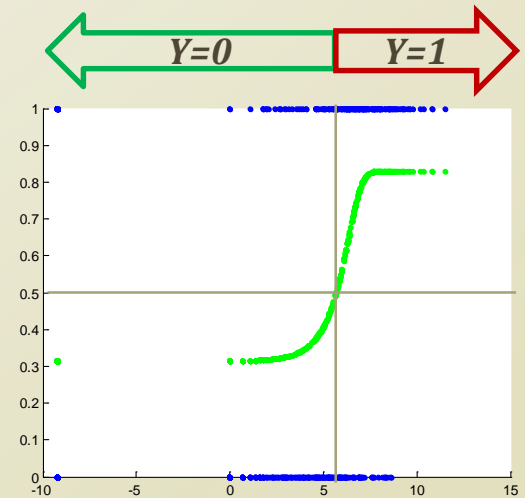
Logit  $\rightarrow$  Logistic  
Inverse of X and Y  
X in Logit is the probability

Linear shift for a better  
function fitting

- When we are fitting the linear regression to approximate  $P(Y|X)$ 
  - $X\theta = P(Y|X)$
  - Though, this is not going to keep the probability axiom
- Now we are fitting to the logistic function to approximate  $P(Y|X)$ 
  - $X\theta = \log\left(\frac{P(Y|X)}{1-P(Y|X)}\right)$
  - From linear to logistic

# Logistic Regression

- Logistic regression is a probabilistic classifier to predict the binomial or the multinomial outcome
  - by fitting the conditional probability to the logistic function.
- You can see the problem from the different view.
  - This way is actually closer to the formal definition.
- Given the Bernoulli experiment
  - $P(y|x) = \mu(x)^y (1 - \mu(x))^{1-y}$
  - $\mu(x) = \frac{1}{1+e^{-\theta^T x}} = P(y = 1|x)$
  - Here,  $\mu(x)$  is the logistic function
- From the previous slide,
  - $X\theta = \log\left(\frac{P(Y|X)}{1-P(Y|X)}\right) \rightarrow P(Y|X) = \frac{e^{X\theta}}{1+e^{X\theta}}$



## Logistic Function

$$f(x) = \frac{1}{1 + e^{-x}}$$

The goal, finally, becomes finding out  $\theta$ , again

$$P(y = 1|x) = \mu(x) = \frac{1}{1 + e^{-\theta^T x}} = \frac{e^{X\theta}}{1 + e^{X\theta}}$$

# Finding the Parameter, $\theta$

$$X\theta = \log\left(\frac{P(Y|X)}{1 - P(Y|X)}\right)$$

- **Maximum Likelihood Estimation (MLE) of  $\theta$**

- Choose  $\theta$  that maximizes the probability of observed data

$$\hat{\theta} = \operatorname{argmax}_{\theta} P(D|\theta)$$

- **This is Maximum Conditional Likelihood Estimation (MCLE)**

- $$\hat{\theta} = \operatorname{argmax}_{\theta} P(D|\theta) = \operatorname{argmax}_{\theta} \prod_{1 \leq i \leq N} P(Y_i|X_i; \theta)$$

$$= \operatorname{argmax}_{\theta} \log\left(\prod_{1 \leq i \leq N} P(Y_i|X_i; \theta)\right) = \operatorname{argmax}_{\theta} \sum_{1 \leq i \leq N} \log(P(Y_i|X_i; \theta))$$

- $$P(Y_i|X_i; \theta) = \mu(X_i)^{Y_i} (1 - \mu(X_i))^{1-Y_i}$$

- $$\begin{aligned} \log(P(Y_i|X_i; \theta)) &= Y_i \log(\mu(X_i)) + (1 - Y_i) \log(1 - \mu(X_i)) \\ &= Y_i \{\log(\mu(X_i)) - \log(1 - \mu(X_i))\} + \log(1 - \mu(X_i)) \\ &= Y_i \log\left(\frac{\mu(X_i)}{1 - \mu(X_i)}\right) + \log(1 - \mu(X_i)) \\ &= Y_i X_i \theta + \log(1 - \mu(X_i)) = Y_i X_i \theta - \log(1 + e^{X_i \theta}) \end{aligned}$$

# Finding the Parameter, $\theta$ , contd.

## Linear Regression (Closed Form):

$$\begin{aligned}\hat{f} &= X\theta & \nabla_{\theta}(\theta^T X^T X \theta - 2\theta^T X^T Y) &= 0 \\ & & 2X^T X \theta - 2X^T Y &= 0 \\ & & \theta &= (X^T X)^{-1} X^T Y\end{aligned}$$

- $\hat{\theta} = \operatorname{argmax}_{\theta} \sum_{1 \leq i \leq N} \log(P(Y_i | X_i; \theta))$
- $= \operatorname{argmax}_{\theta} \sum_{1 \leq i \leq N} \{Y_i X_i \theta - \log(1 + e^{X_i \theta})\}$
- Partial derivative to find a certain element in  $\theta$

$$\frac{\partial}{\partial \theta_j} \left\{ \sum_{1 \leq i \leq N} Y_i X_i \theta - \log(1 + e^{X_i \theta}) \right\} \quad P(y = 1|x) = \frac{e^{x\theta}}{1 + e^{x\theta}}$$

$$= \left\{ \sum_{1 \leq i \leq N} Y_i X_{i,j} \right\} + \left\{ \sum_{1 \leq i \leq N} -\frac{1}{1 + e^{X_i \theta}} \times e^{X_i \theta} \times X_{i,j} \right\}$$

$$= \sum_{1 \leq i \leq N} X_{i,j} \left( Y_i - \frac{e^{X_i \theta}}{1 + e^{X_i \theta}} \right) = \sum_{1 \leq i \leq N} X_{i,j} (Y_i - P(Y_i = 1 | X_i; \theta)) = 0$$

- There is no way to derive further
  - There is no closed form solution!
  - Open form solution  $\rightarrow$  approximate!

Cannot be easily solved in the closed form because of the logistic function

# GRADIENT METHOD

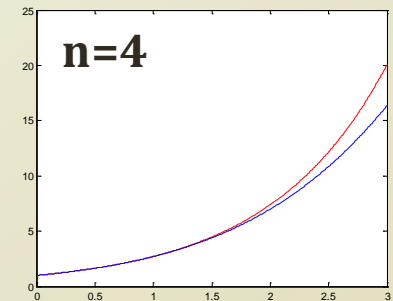
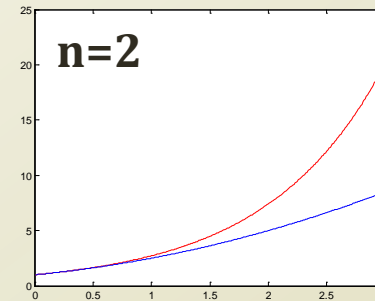
# Taylor Expansion

- Taylor series is a representation of a function
  - as a infinite sum of terms calculated from the values of the function's derivatives at a fixed point.
- $$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \dots$$

$$= \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$
  - $a$  = a constant value
- Taylor series is possible when
  - Infinitely differentiable at a real or complex number of  $a$
- Taylor expansion is a process of generating the Taylor series

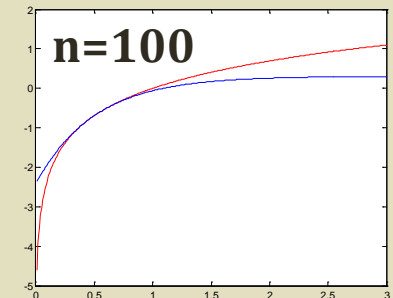
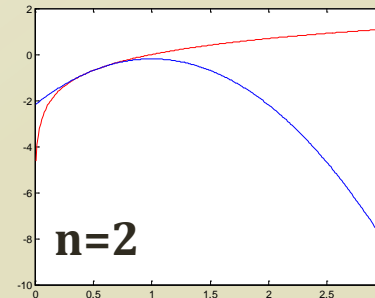
when  $a = 0$ ,

$$e^x = 1 + \frac{e^0}{1!}(x-0)^1 + \frac{e^0}{2!}(x-0)^2 + \dots$$



when  $a = 0.5$ ,

$$\log x = \log(0.5) + \frac{0.5}{1!}(x-0.5)^1 + \frac{1}{\frac{0.5^2}{2!}}(x-0.5)^2 + \dots$$



# Gradient Descent/Ascent

- Gradient descent/ascent method is
  - Given a differentiable function of  $f(x)$  and an initial parameter of  $x_1$
  - Iteratively moving the parameter to the lower/higher value of  $f(x)$
  - By taking the direction of the negative/positive gradient of  $f(x)$

- Why this works?

- $f(x) = f(a) + \frac{f'(a)}{1!}(x - a) + O(\|x - a\|^2)$  Useful Big-Oh Notation

- Assume  $a=x_1$  and  $x=x_1+h\mathbf{u}$ ,  $\mathbf{u}$  is the unit direction vector for the partial deriv.

- $f(x_1 + h\mathbf{u}) = f(x_1) + hf'(x_1)\mathbf{u} + h^2O(1)$

- $f(x_1 + h\mathbf{u}) - f(x_1) \approx hf'(x_1)\mathbf{u}$

Always???

- $\mathbf{u}^* = \operatorname{argmin}_{\mathbf{u}} \{f(x_1 + h\mathbf{u}) - f(x_1)\} = \operatorname{argmin}_{\mathbf{u}} hf'(x_1)\mathbf{u} = -\frac{f'(x_1)}{|f'(x_1)|}$

- $\because f(x_1 + h\mathbf{u}) \leq f(x_1), \vec{a} \cdot \vec{b} = |\vec{a}||\vec{b}|\cos\alpha$

Gradient Descent

- $x_{t+1} \leftarrow x_t + h\mathbf{u}^* = x_t - h\frac{f'(x_t)}{|f'(x_t)|}$

- Perfectly applicable to  $\hat{\theta} = \operatorname{argmax}_{\theta} \sum_{1 \leq i \leq N} \log(P(Y_i|X_i; \theta))$

- $f(\theta) = \sum_{1 \leq i \leq N} \log(P(Y_i|X_i; \theta))$

- Setup an initial parameter of  $\theta_1$

- Iteratively moving  $\theta_t$  to the higher value of  $f(\theta_t)$

- By taking the direction of the **positive** gradient of  $f(\theta_t)$

Gradient Ascent



# How Gradient Descent Works

- Example function: Rosenbrock function

- $f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$
- $\frac{\partial}{\partial x_1} f(x_1, x_2) = -2(1 - x_1) - 400x_1(x_2 - x_1^2)$
- $\frac{\partial}{\partial x_2} f(x_1, x_2) = 200(x_2 - x_1^2)$

- Assume the initial point

- $\mathbf{x}^0 = (x_1^0, x_2^0) = (-1.3, 0.9)$

Global Minimum=0  
at (1,1)

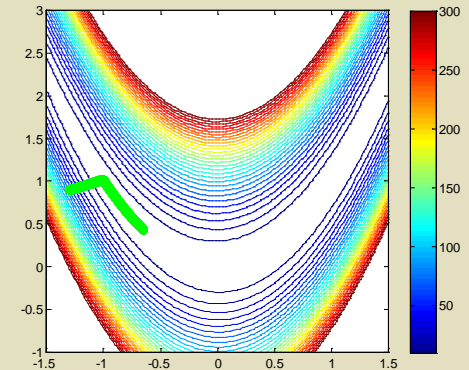
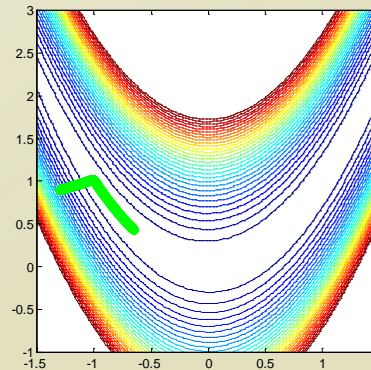
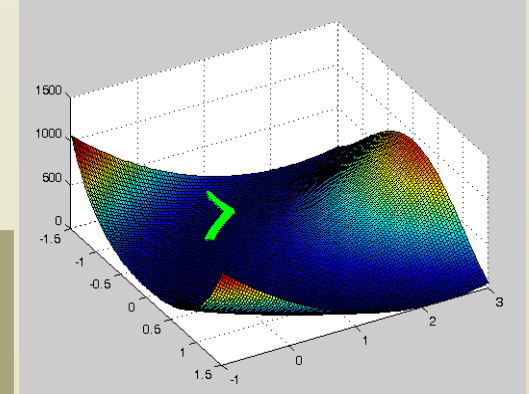
- Partial derivative vector at the point

- $\mathbf{f}'(\mathbf{x}^0) = \left( \frac{\partial}{\partial x_1} f(x_1, x_2), \frac{\partial}{\partial x_2} f(x_1, x_2) \right) = (-415.4, -158)$

- Update the point with the negative partial derivative in a small scale,  
 $h=0.001$

- $\mathbf{x}^1 \leftarrow \mathbf{x}^0 - h \frac{\mathbf{f}'(\mathbf{x}^0)}{|\mathbf{f}'(\mathbf{x}^0)|}$
- $\mathbf{x}^1 = \begin{pmatrix} -1.3 - 0.001 \times -415.4/444.4335, \\ 0.9 - 0.001 \times -158/444.4335 \end{pmatrix}$
- $= (-1.2991, 0.9004)$

- Repeat the update until converges





$$P(y = 1|x) = \frac{e^{x\theta}}{1 + e^{x\theta}}$$

# Finding $\theta$ with Gradient Ascent

- $\hat{\theta} = \operatorname{argmax}_{\theta} \sum_{1 \leq i \leq N} \log(P(Y_i|X_i; \theta))$ 
  - $f(\theta) = \sum_{1 \leq i \leq N} \log(P(Y_i|X_i; \theta))$
  - $\frac{\partial f(\theta)}{\partial \theta_j} = \frac{\partial}{\partial \theta_j} \{ \sum_{1 \leq i \leq N} \log(P(Y_i|X_i; \theta)) \} = \sum_{1 \leq i \leq N} X_{i,j} (Y_i - P(y = 1|x; \theta))$
- To utilize the gradient method
  - We need to know  $f'(x)$  which are above
    - Case of ascent:  $x_{t+1} \leftarrow x_t + h\mathbf{u}^* = x_t + h \frac{f'(x_t)}{|f'(x_t)|}$
  - Then, how to iteratively update the parameter,  $\theta$
  - $\theta_j^{t+1} \leftarrow \theta_j^t + h \frac{\partial f(\theta^t)}{\partial \theta_j^t} = \theta_j^t + h \{ \sum_{1 \leq i \leq N} X_{i,j} (Y_i - P(Y = 1|X_i; \theta^t)) \}$ 


$$= \theta_j^t + \frac{h}{C} \left\{ \sum_{1 \leq i \leq N} X_{i,j} \left( Y_i - \frac{e^{X_i \theta^t}}{1 + e^{X_i \theta^t}} \right) \right\}$$
  - $\theta_j^0$  can be arbitrarily chosen.

C=Normalization to the unit vector

# Logistic Regression Matlab Exercise

- Let's do some coding...

# Linear Regression Revisited

- Previously,
  - $\hat{\theta} = \operatorname{argmin}_{\theta} (f - \hat{f})^2 = \operatorname{argmin}_{\theta} (Y - X\theta)^2$   
 $= \operatorname{argmin}_{\theta} (Y - X\theta)^T (Y - X\theta) = \operatorname{argmin}_{\theta} (Y - X\theta)^T (Y - X\theta)$   
 $= \operatorname{argmin}_{\theta} (\theta^T X^T X \theta - 2\theta^T X^T Y + Y^T Y) = \operatorname{argmin}_{\theta} (\theta^T X^T X \theta - 2\theta^T X^T Y)$
  - $\nabla_{\theta} (\theta^T X^T X \theta - 2\theta^T X^T Y) = 0$ 
    - $2X^T X \theta - 2X^T Y = 0$
  - $\theta = (X^T X)^{-1} X^T Y$
- Any problem??? 
- Gradient descent can be a solution
  - $\hat{\theta} = \operatorname{argmin}_{\theta} (f - \hat{f})^2 = \operatorname{argmin}_{\theta} (Y - X\theta)^2 =$   
 $\operatorname{argmin}_{\theta} \sum_{1 \leq i \leq N} (Y^i - \sum_{1 \leq j \leq d} X_j^i \theta_j)^2$
  - $\frac{\partial}{\partial \theta_k} \sum_{1 \leq i \leq N} (Y^i - \sum_{1 \leq j \leq d} X_j^i \theta_j)^2 = - \sum_{1 \leq i \leq N} 2(Y^i - \sum_{1 \leq j \leq d} X_j^i \theta_j) X_k^i$
  - $\theta_k^{t+1} \leftarrow \theta_k^t - h \frac{\partial f(\theta^t)}{\partial \theta_k^t} = \theta_k^t + h \sum_{1 \leq i \leq N} 2(Y^i - \sum_{1 \leq j \leq d} X_j^i \theta_j) X_k^i$

# NAÏVE BAYES VS. LOGISTIC REGRESSION

# Gaussian Naïve Bayes

- We want to compare the performance of the two classifiers
  - Logistic regression handles the continuous features
  - Why not naïve Bayes?
- Naïve Bayes Classifier Function
  - $f_{NB}(x) = \operatorname{argmax}_{Y=y} P(Y = y) \prod_{1 \leq i \leq d} P(X_i = x_i | Y = y)$
- What-if the feature is a continuous random variable?
  - We can assume that the variable follows the Gaussian distribution with the mean of  $\mu$  and the variance of  $\sigma^2$ 
    - $P(X_i | Y, \mu, \sigma^2) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(X_i - \mu)^2}{2\sigma^2}}$
  - In addition, let's use more shortened terms
    - $P(Y = y) = \pi_1$
  - $P(Y) \prod_{1 \leq i \leq d} P(X_i | Y) = \pi_k \prod_{1 \leq i \leq d} \frac{1}{\sigma_k^i C} \exp\left(-\frac{1}{2} \left(\frac{X_i - \mu_k^i}{\sigma_k^i}\right)^2\right)$

# Derivation to Logistic Regression (1)

- Derivation from the naïve Bayes to the logistic regression

- $$P(Y) \prod_{1 \leq i \leq d} P(X_i|Y) = \pi_k \prod_{1 \leq i \leq d} \frac{1}{\sigma_k^i C} \exp\left(-\frac{1}{2} \left(\frac{X_i - \mu_k^i}{\sigma_k^i}\right)^2\right)$$

- With naïve Bayes assumption

- $$P(Y = y|X) = \frac{P(X|Y = y)P(Y=y)}{P(X)} = \frac{P(X|Y = y)P(Y=y)}{P(X|Y = y)P(Y=y) + P(X|Y = n)P(Y=n)}$$

$$= \frac{P(Y = y) \prod_{1 \leq i \leq d} P(X_i|Y = y)}{P(Y = y) \prod_{1 \leq i \leq d} P(X_i|Y = y) + P(Y = n) \prod_{1 \leq i \leq d} P(X_i|Y = n)}$$

- $$P(Y = y|X) = \frac{\pi_1 \prod_{1 \leq i \leq d} \frac{1}{\sigma_1^i C} \exp\left(-\frac{1}{2} \left(\frac{X_i - \mu_1^i}{\sigma_1^i}\right)^2\right)}{\pi_1 \prod_{1 \leq i \leq d} \frac{1}{\sigma_1^i C} \exp\left(-\frac{1}{2} \left(\frac{X_i - \mu_1^i}{\sigma_1^i}\right)^2\right) + \pi_2 \prod_{1 \leq i \leq d} \frac{1}{\sigma_2^i C} \exp\left(-\frac{1}{2} \left(\frac{X_i - \mu_2^i}{\sigma_2^i}\right)^2\right)}$$

$$= \frac{1}{1 + \frac{\pi_2 \prod_{1 \leq i \leq d} \frac{1}{\sigma_2^i C} \exp\left(-\frac{1}{2} \left(\frac{X_i - \mu_2^i}{\sigma_2^i}\right)^2\right)}{\pi_1 \prod_{1 \leq i \leq d} \frac{1}{\sigma_1^i C} \exp\left(-\frac{1}{2} \left(\frac{X_i - \mu_1^i}{\sigma_1^i}\right)^2\right)}}$$

# Derivation to Logistic Regression (2)

- Assuming the same variable of the two classes,  $\sigma_2^i = \sigma_1^i$

$$\begin{aligned}
 P(Y = y|X) &= \frac{1}{\pi_2 \prod_{1 \leq i \leq d} \frac{1}{\sigma_2^i C} \exp(-\frac{1}{2}(\frac{X_i - \mu_2^i}{\sigma_2^i})^2)} = \frac{1}{\pi_2 \prod_{1 \leq i \leq d} \exp(-\frac{1}{2}(\frac{X_i - \mu_2^i}{\sigma_2^i})^2)} \\
 &\quad \frac{1}{1 + \frac{\pi_1 \prod_{1 \leq i \leq d} \frac{1}{\sigma_1^i C} \exp(-\frac{1}{2}(\frac{X_i - \mu_1^i}{\sigma_1^i})^2)}{\pi_2 \prod_{1 \leq i \leq d} \frac{1}{\sigma_2^i C} \exp(-\frac{1}{2}(\frac{X_i - \mu_2^i}{\sigma_2^i})^2)}} = \frac{1}{1 + \frac{\pi_1 \prod_{1 \leq i \leq d} \exp(-\frac{1}{2}(\frac{X_i - \mu_1^i}{\sigma_1^i})^2)}{\pi_2 \prod_{1 \leq i \leq d} \exp(-\frac{1}{2}(\frac{X_i - \mu_2^i}{\sigma_2^i})^2)}} \\
 &= \frac{1}{1 + \frac{\pi_1 \exp(-\sum_{1 \leq i \leq d} \{\frac{1}{2}(\frac{X_i - \mu_1^i}{\sigma_1^i})^2\})}{\pi_2 \exp(-\sum_{1 \leq i \leq d} \{\frac{1}{2}(\frac{X_i - \mu_2^i}{\sigma_2^i})^2\})}} = \frac{1}{1 + \frac{\exp(-\sum_{1 \leq i \leq d} \{\frac{1}{2}(\frac{X_i - \mu_2^i}{\sigma_2^i})^2\} + \log \pi_2)}{\exp(-\sum_{1 \leq i \leq d} \{\frac{1}{2}(\frac{X_i - \mu_1^i}{\sigma_1^i})^2\} + \log \pi_1)}} \\
 &= \frac{1}{1 + \exp(-\sum_{1 \leq i \leq d} \{\frac{1}{2}(\frac{X_i - \mu_2^i}{\sigma_2^i})^2\} + \log \pi_2 + \sum_{1 \leq i \leq d} \{\frac{1}{2}(\frac{X_i - \mu_1^i}{\sigma_1^i})^2\} - \log \pi_1)} \\
 &= \frac{1}{1 + \exp(\sum_{1 \leq i \leq d} -\frac{1}{2(\sigma_1^i)^2} (\{(X_i - \mu_1^i)^2\} - \{(X_i - \mu_2^i)^2\}) + \log \pi_2 - \log \pi_1)} \\
 &= \frac{1}{1 + \exp(\sum_{1 \leq i \leq d} -\frac{1}{2(\sigma_1^i)^2} (\{2(\mu_2^i - \mu_1^i)X_i + \mu_2^{i2} - \mu_1^{i2}\} + \log \pi_2 - \log \pi_1))}
 \end{aligned}$$

# Naïve Bayes vs. Logistic Regression

- Naïve Bayes classifier

- $$P(Y|X) = \frac{1}{1 + \exp\left(-\frac{1}{2(\sigma_1^i)^2} \sum_{1 \leq i \leq d} \{2(\mu_2^i - \mu_1^i)X_i + \mu_2^{i2} - \mu_1^{i2}\} + \log \pi_2 - \log \pi_1\right)}$$

- Assumption to get this formula

- Naïve Bayes assumption, Same variance assumption between classes
- Gaussian distribution for  $P(X|Y)$
- Bernoulli distribution for  $P(Y)$

Together, modeling joint prob.

- # of parameters to estimate =  $2 \times 2 \times d + 1 = 4d + 1$

- With the different variances between classes

- Logistic Regression

- $$P(Y|X) = \frac{1}{1 + e^{-\theta^T x}}$$

- Assumption to get this formula

- Fitting to the logistic function

- # of parameters to estimate =  $d + 1$

- Who is the winner?

- Really??? There is no winner... Why?



# Generative-Discriminative Pair

- Generative model,  $P(Y|X)=P(X,Y)/P(X)=P(X|Y)P(Y)/P(X)$ 
  - Full probabilistic model of all variables
    - Estimate the parameters of  $P(X|Y)$ ,  $P(Y)$  from the data
  - Characteristics: Bayesian, Prior, Modeling the joint probability
  - Naïve Bayes Classifier
- Discriminative model,  $P(Y|X)$ 
  - Do not need to model the distribution of the observed variables
    - Estimate the parameters of  $P(Y|X)$  from the data
  - Characteristics: Modeling the conditional probability
  - Logistic Regression
- Pros and Cons [Ng & Jordan, 2002]
  - Logistic regression is less biased
  - Probably approximately correct learning: Naïve Bayes learns faster

# Acknowledgement

- This slideset is greatly influenced
  - By Prof. Carlos Guestrin at CMU
  - By Prof. Eric Xing at CMU
  - By Prof. Tom Mitchell at CMU

# Further Readings

- Bishop Chapter 4.3, 5.2.1-5.2.4