CS6220: DATA MINING TECHNIQUES

Matrix Data: Classification: Part 2

Instructor: Yizhou Sun

yzsun@ccs.neu.edu

February 3, 2016

Methods to Learn

	Matrix Data	Text Data	Set Data	Sequence Data	Time Series	Graph & Network	Images

Classification	Decision Tree;
	Naïve Bayes;
	Logistic Regression
	SVM; kNN
				HMM		Label Propagation	Neural Network

Clustering	K-means; hierarchical clustering; DBSCAN; Mixture Models; kernel k-means*	PLSA				SCAN; Spectral Clustering	
Frequent Pattern Mining			Apriori; FP-growth	GSP; PrefixSpan			
Prediction	Linear Regression				Autoregression	Collaborative Filtering	
Similarity Search					DTW	P-PageRank	
Ranking						PageRank	

Matrix Data: Classification: Part 2

- Bayesian Learning
  - Naïve Bayes
  - Bayesian Belief Network
- Logistic Regression
- Summary

Basic Probability Review

- Have two dice h_1 and h_2
- The probability of rolling an *i* given die h_1 is denoted $P(i|h_1)$. This is a *conditional probability*
- Pick a die at random with probability $P(h_j)$, j=1 or 2. The probability for picking die h_j and rolling an i with it is called *joint probability* and is $P(i, h_j)=P(h_j)P(i| h_j)$.
- If we know $P(i| h_j)$, then the so-called *marginal probability* P(i) can be computed as: $P(i) = \sum_j P(i, h_j)$
- For any X and Y, $P(X,Y)=P(X|Y)P(Y)$

Bayes' Theorem: Basics

- Bayes' Theorem: $$P(h|\mathbf{X}) = \frac{P(\mathbf{X}|h)P(h)}{P(\mathbf{X})}$$
  - Let **X** be a data sample ("*evidence*")
  - Let h be a *hypothesis* that X belongs to class C
  - P(h) (*prior probability*): the initial probability
    - E.g., **X** will buy computer, regardless of age, income, …
  - P(**X**|h) (likelihood): the probability of observing the sample **X**, given that the hypothesis holds
    - E.g., Given that **X** will buy computer, the prob. that X is 31..40, medium income
  - P(**X**): marginal probability that sample data is observed
    - $P(X) = \sum_h P(X|h)\,P(h)$
  - P(h|**X**), (i.e., *posterior probability):* the probability that the hypothesis holds given the observed data sample **X**

Classification: Choosing Hypotheses

- *Maximum Likelihood* (maximize the likelihood):

$$h_{ML} = \arg\max_{h \in H} P(X \mid h)$$

- *Maximum a posteriori* (maximize the posterior):
  - Useful observation: it does not depend on the denominator P(X)

$$h_{MAP} = \arg\max_{h \in H} P(h \mid X) = \arg\max_{h \in H} P(X \mid h) P(h)$$

Classification by Maximum A Posteriori

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an p-D attribute vector **X** = (x_1, x_2, …, x_p)
- Suppose there are *m* classes Y∈{C_1, C_2, …, C_m}
- Classification is to derive the maximum posteriori, i.e., the maximal P(Y=C_j|**X**)
- This can be derived from Bayes' theorem $P(Y=C_j|\mathbf{X})=\frac{P(\mathbf{X}|Y=C_j)P(Y=C_j)}{P(\mathbf{X})}$
- Since P(X) is constant for all classes, only $P(y,\mathbf{X})=P(\mathbf{X}|y)P(y)$ needs to be maximized

Example: Cancer Diagnosis

- A patient takes a lab test with two possible results (+ve, -ve), and the result comes back positive. It is known that the test returns
  - a correct positive result in only 98% of the cases;
  - a correct negative result in only 97% of the cases.
  - Furthermore, only 0.008 of the entire population has this disease.

1. What is the probability that this patient has cancer?
2. What is the probability that he does not have cancer?
3. What is the diagnosis?

Solution

P(cancer) = .008 P(¬ cancer) = .992

P(+ve|cancer) = .98 P(-ve|cancer) = .02

P(+ve| ¬ cancer) = .03 P(-ve| ¬ cancer) = .97

Using Bayes Formula:

P(cancer|+ve) = P(+ve|cancer)xP(cancer) / P(+ve)

= 0.98 x 0.008/ P(+ve) = .00784 / P(+ve)

P(¬ cancer|+ve) = P(+ve| ¬ cancer)xP(¬ cancer) / P(+ve)

= 0.03 x 0.992/P(+ve) = .0298 / P(+ve)

So, the patient most likely does not have cancer.

Matrix Data: Classification: Part 2

- Bayesian Learning
  - Naïve Bayes
  - Bayesian Belief Network
- Logistic Regression
- Summary

Naïve Bayes Classifier

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an p-D attribute vector **X** = (x_1, x_2, …, x_p)
- Suppose there are *m* classes Y∈{C_1, C_2, …, C_m}
- Goal: Find Y $\max P(Y|\boldsymbol{X}) = P(Y,\boldsymbol{X})/P(\boldsymbol{X}) \propto P(\boldsymbol{X}|Y)P(Y)$
- A simplified assumption: attributes are conditionally independent given the class (class conditional independency):

$$P(\mathbf{X}|C_j) = \prod_{k=1}^{p} P(x_k|C_j) = P(x_1|C_j) \times P(x_2|C_j) \times \ldots \times P(x_p|C_j)$$

Estimate Parameters by MLE

- Given a dataset $D = \{(\mathbf{X}_\mathrm{i}, \mathrm{Y}_\mathrm{i})\}$, the goal is to
  - Find the best estimators $P(C_j)$ and $P(X_k = x_k|C_j)$, for every $j = 1, \ldots, m \ and \ k = 1, \ldots, p$
  - that maximizes the likelihood of observing D:

$$L = \prod_i P(\mathbf{X}_\mathrm{i}, \mathrm{Y}_\mathrm{i}) = \prod_i P(\mathbf{X}_\mathrm{i}|\mathrm{Y}_\mathrm{i})P(Y_i)$$

$$= \prod_i (\prod_k P(X_{ik}|Y_i))P(Y_i)$$

- Estimators of Parameters:
  - $P(C_j) = |C_{j,D}|/|D|$ ($|C_{j,D}|$= # of tuples of C_j in D) (why?)
  - $P(X_k = x_k|C_j)$: X_k can be either discrete or numerical

Discrete and Continuous Attributes

- If X_k is discrete, with V possible values
  - P(x_k|C_j) is the # of tuples in C_j having value x_k for X_k divided by |$C_{j, D}$|
- If X_k is continuous, with observations of real values
  - P(x_k|C_j) is usually computed based on Gaussian distribution with a mean μ and standard deviation σ
  - Estimate (μ, σ^2) according to the observed X in the category of C_j
    - Sample mean and sample variance
  - P(x_k|C_j) is then $P(X_k = x_k \mid C_j) = g(x_k, \mu_{C_j}, \sigma_{C_j})$

Gaussian density function

Naïve Bayes Classifier: Training Dataset

Class:

C1:buys_computer = ‘yes’

C2:buys_computer = ‘no’

Data to be classified:

X = (age <=30,

Income = medium,

Student = yes

Credit_rating = Fair)

age	income	student	credit_rating	_comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Naïve Bayes Classifier: An Example

age	income	student	credit_rating	_comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

- $P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357
- Compute $P(X|C_i)$ for each class
  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4
- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

$P(X|C_i)$: P(**X**|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(**X**|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

$P(X|C_i)*P(C_i)$: P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028
P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")

Avoiding the Zero-Probability Problem

- Naïve Bayesian prediction requires each conditional prob. be **non-zero**. Otherwise, the predicted prob. will be zero

$$P(X|C_j) = \prod_{k=1}^{p} P(x_k|C_j)$$

- Use **Laplacian correction** (or Laplacian smoothing)
  - *Adding 1 to each case*
    - $P(x_k = v|C_j) = \frac{n_{jk,v}+1}{|C_{j,D}|+V}$ where $n_{jk,v}$ is # of tuples in C_j having value $x_k = \text{v}$, V is the total number of values that can be taken
    - Ex. Suppose a training dataset with 1000 tuples, for category "buys_computer = yes", income=low (0), income= medium (990), and income = high (10)

      Prob(income = low|buys_computer = "yes") = 1/1003

      Prob(income = medium|buys_computer = "yes") = 991/1003

      Prob(income = high|buys_computer = "yes") = 11/1003

  - The "corrected" prob. estimates are close to their "uncorrected" counterparts

*Smoothing and Prior on Attribute Distribution

- $Discrete\ distribution$: $X_k | C_j \sim \boldsymbol{\theta}$
  - $P\left(X_k = v \middle| C_j, \boldsymbol{\theta}\right) = \theta_v$
- Put prior to $\boldsymbol{\theta}$
  - In discrete case, the prior can be chosen as symmetric Dirichlet distribution: $\boldsymbol{\theta} \sim Dir(\alpha),\ i.e., P(\boldsymbol{\theta}) \propto \prod_v \theta_v^{\alpha-1}$
  - $posterior\ distribution$:
    - $P\left(\theta \middle| X_{1k}, \ldots, X_{nk}, C_j\right) \propto P\left(X_{1k}, \ldots, X_{nk} \middle| C_j, \boldsymbol{\theta}\right) P(\boldsymbol{\theta})$, another Dirichlet distribution, with new parameter $(\alpha + c_1, \ldots, \alpha + c_v, \ldots, \alpha + c_V)$
    - c_v is the number of observations taking value v
  - Inference: $P\left(X_k = v \middle| X_{1k}, \ldots, X_{nk}, C_j\right) = \int P(X_k = v | \boldsymbol{\theta}) P\left(\boldsymbol{\theta} \middle| X_{1k}, \ldots, X_{nk}, C_j\right) \mathrm{d}\boldsymbol{\theta}$

$$= \frac{\boldsymbol{c_v + \alpha}}{\boldsymbol{\sum c_v + V\alpha}}$$

    - Equivalent to adding α to each observation value v

*Notes on Parameter Learning

- Why the probability of $P\left(X_k \middle| C_j\right)$ is estimated in this way?
  - http://www.cs.columbia.edu/~mcollins/em.pdf
  - http://www.cs.ubc.ca/~murphyk/Teaching/CS340-Fall06/reading/NB.pdf

Naïve Bayes Classifier: Comments

- Advantages
  - Easy to implement
  - Good results obtained in most of the cases
- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., hospitals: patients: Profile: age, family history, etc.

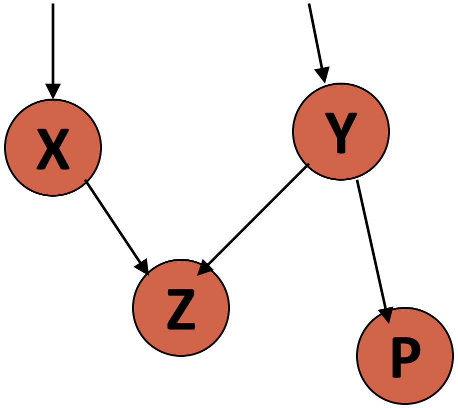
      Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayes Classifier
- How to deal with these dependencies? Bayesian Belief Networks

Matrix Data: Classification: Part 2

- Bayesian Learning
  - Naïve Bayes
  - Bayesian Belief Network
- Logistic Regression
- Summary

Bayesian Belief Networks (BNs)

- **Bayesian belief network** (also known as **Bayesian network**, **probabilistic network**): allows *class conditional independencies* between *subsets* of variables
- Two components: (1) A *directed acyclic graph* (called a structure) and (2) a set of *conditional probability tables* (CPTs)
- A (*directed acyclic*) graphical model of *causal influence* relationships
  - Represents dependency among the variables
  - Gives a specification of joint probability distribution



❑ Nodes: random variables

❑ Links: dependency

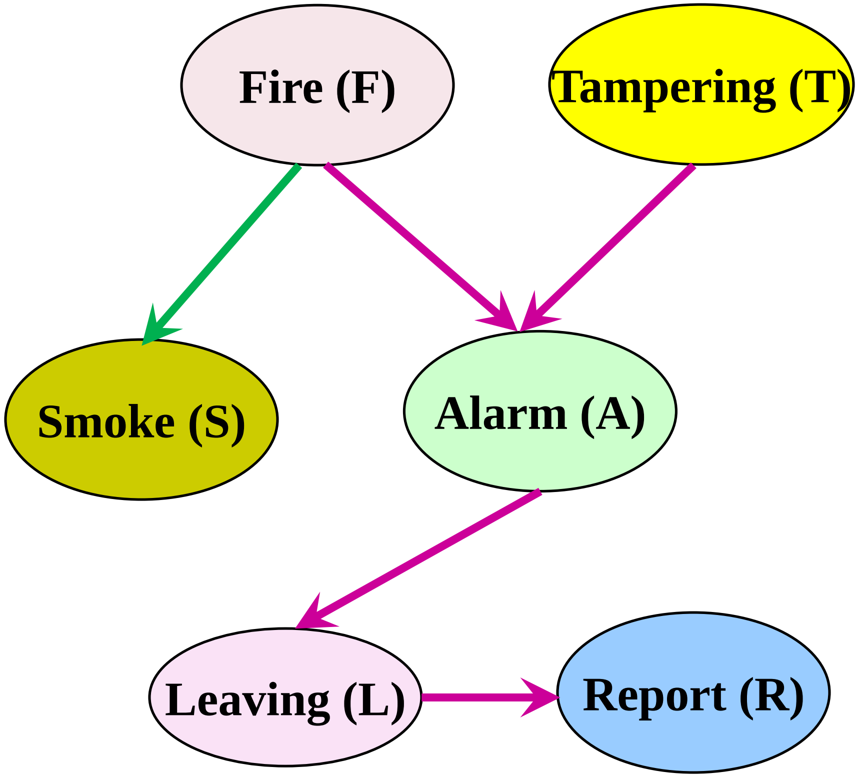
❑ X and Y are the parents of Z, and Y is the parent of P

❑ No dependency between Z and P conditional on Y

❑ Has no cycles

21

A Bayesian Network and Some of Its CPTs



CPT: Conditional Probability Tables

	F	¬F
S	.90	.01
¬S	.10	.99

	F, T	*F*, ¬*T*	¬*F, T*	¬*F*, ¬*T*
A	.5	.99	.85	.0001
¬A	.95	.01	.15	.9999

CPT shows the conditional probability for each possible combination of its parents

Derivation of the probability of a particular combination of values of **X**, from CPT (joint probability):

$$P(x_1,\ldots,x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(x_i))$$

Inference in Bayesian Networks

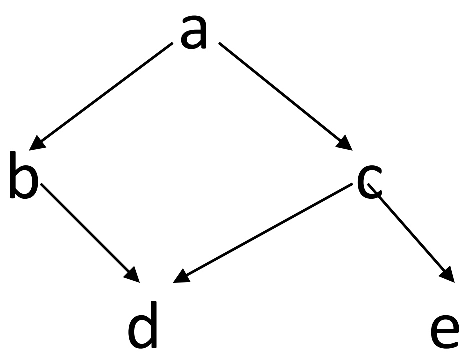
- Infer the probability of values of some variable given the observations of other variables
  - E.g., P(Fire = True|Report = True, Smoke = True)?
- Computation
  - Exact computation by enumeration
  - In general, the problem is NP hard
    - *Approximation algorithms are needed

Inference by enumeration

- To compute posterior marginal $P(X_i \mid E=e)$
  - Add all of the terms (atomic event probabilities) from the full joint distribution
  - If **E** are the evidence (observed) variables and **Y** are the other (unobserved) variables, then:

    $P(X|\mathbf{e}) = \alpha\, P(X, \mathbf{E}) = \alpha \sum P(X, \mathbf{E}, \mathbf{Y})$
  - Each P(X, **E**, **Y**) term can be computed using the chain rule
- Computationally expensive!

Example: Enumeration



- $P\ (d|e) = \alpha\ \Sigma_{ABC}P(a, b, c, d, e)$
  $= \alpha\ \Sigma_{ABC}P(a)\ P(b|a)\ P(c|a)\ P(d|b,c)\ P(e|c)$
- With simple iteration to compute this expression, there’s going to be a lot of repetition (e.g., P(e|c) has to be recomputed every time we iterate over C=true)
  - *A solution: variable elimination

*How Are Bayesian Networks Constructed?

- **Subjective construction**: Identification of (direct) causal structure
  - People are quite good at identifying direct causes from a given set of variables & whether the set contains all relevant direct causes
  - Markovian assumption: Each variable becomes independent of its non-effects once its direct causes are known
  - E.g., S ← F → A ← T, path S→A is blocked once we know F→A
- **Synthesis from other specifications**
  - E.g., from a formal system design: block diagrams & info flow
- **Learning from data**
  - E.g., from medical records or student admission record
  - Learn parameters give its structure or learn both structure and parms
  - Maximum likelihood principle: favors Bayesian networks that maximize the probability of observing the given data set

*Learning Bayesian Networks: Several Scenarios

- Scenario 1: Given both the network structure and all variables observable: *compute only the CPT entries (Easiest case!)*
- Scenario 2: Network structure known, some variables hidden: *gradient descent* (greedy hill-climbing) method, i.e., search for a solution along the steepest descent of a criterion function
  - Weights are initialized to random probability values
  - At each iteration, it moves towards what appears to be the best solution at the moment, w.o. backtracking
  - Weights are updated at each iteration & converge to local optimum
- Scenario 3: Network structure unknown, all variables observable: search through the model space to *reconstruct network topology*
- Scenario 4: Unknown structure, all hidden variables: No good algorithms known for this purpose
- D. Heckerman. A Tutorial on Learning with Bayesian Networks. In *Learning in Graphical Models,* M. Jordan, ed. MIT Press, 1999.

Matrix Data: Classification: Part 2

- Bayesian Learning
  - Naïve Bayes
  - Bayesian Belief Network
- Logistic Regression
- Summary

Linear Regression VS. Logistic Regression

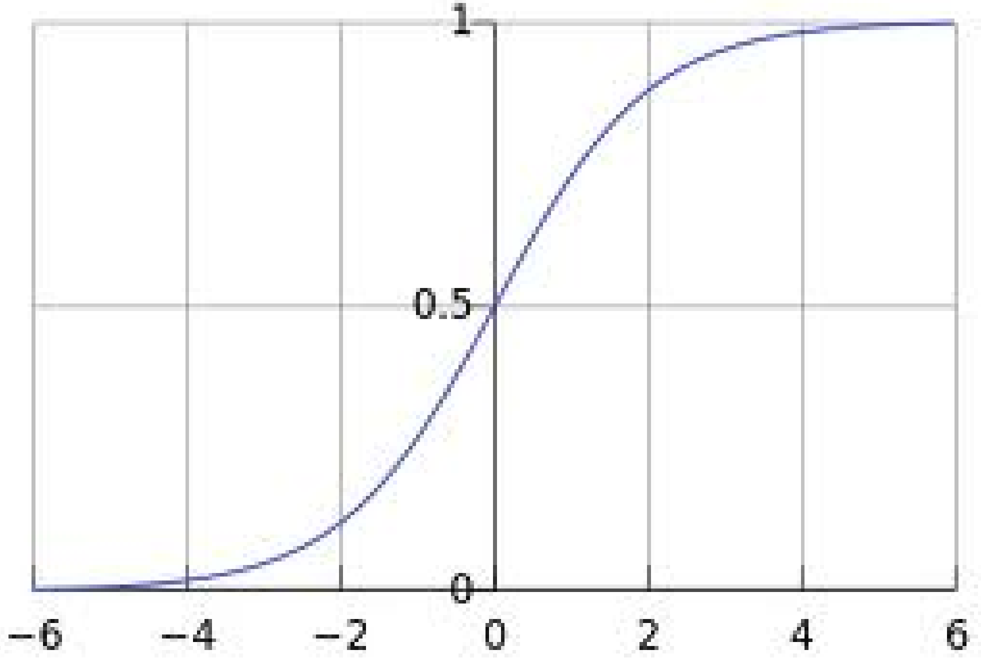
- Linear Regression
  - $\mathrm{Y}: \textit{continuous value } (-\infty, +\infty)$
    - $\mathsf{Y} = \boldsymbol{x}^T\boldsymbol{\beta} = \beta_0 + x_1\beta_1 + x_2\beta_2 + \cdots + x_p\beta_p$
    - $\mathsf{Y}|\boldsymbol{x}, \beta \sim N(\boldsymbol{x}^T\beta, \sigma^2)$

- Logistic Regression
  - $\mathrm{Y}: \textit{discrete value from m classes}$
  - $p\left(Y = C_j\right) \in [0,1] \textit{ and } \sum_j p\left(Y = C_j\right) = 1$

Logistic Function

- Logistic Function / sigmoid function:

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



Modeling Probabilities of Two Classes

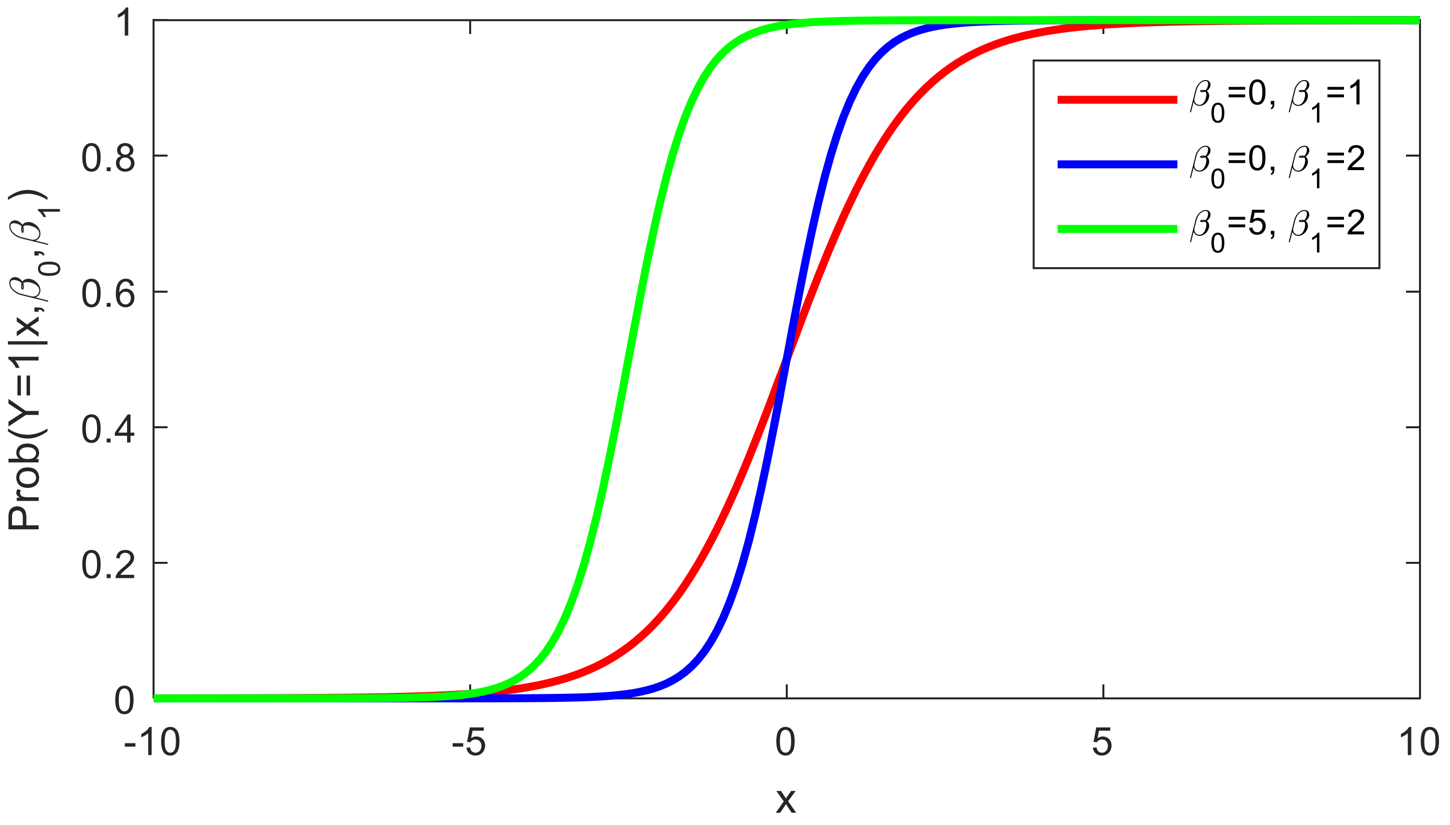
- $P(Y = 1|X,\beta) = \sigma\left(X^T\beta\right) = \frac{1}{1+\exp\{-X^T\beta\}} = \frac{\exp\{X^T\beta\}}{1+\exp\{X^T\beta\}}$
- $P(Y = 0|X,\beta) = 1 - \sigma\left(X^T\beta\right) = \frac{\exp\{-X^T\beta\}}{1+\exp\{-X^T\beta\}} = \frac{1}{1+\exp\{X^T\beta\}}$

$$\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}$$

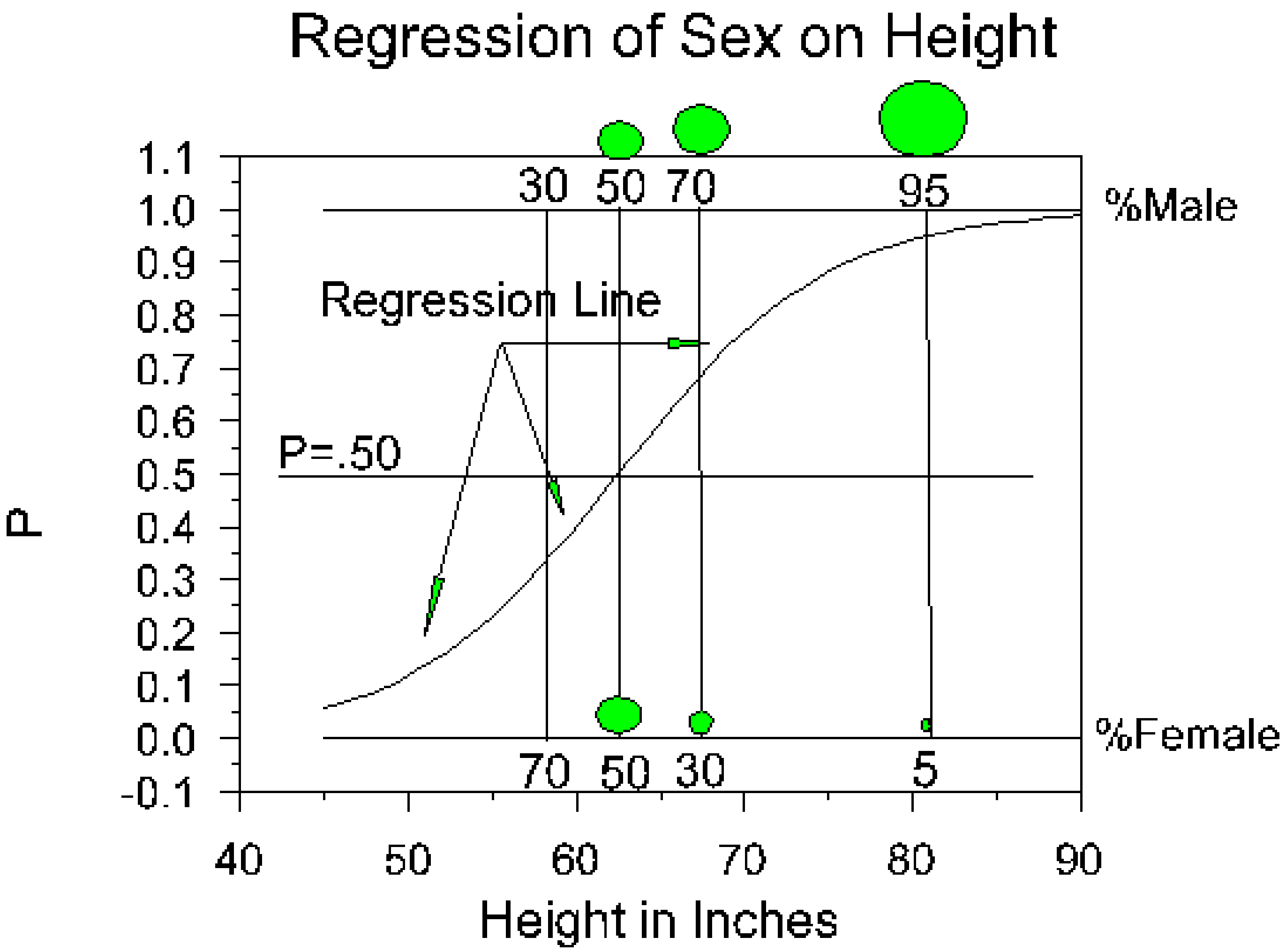
- In other words
  - $Y|\mathrm{X},\beta \sim Bernoulli(\sigma\left(X^T\beta\right))$

The 1-d Situation

- $P(Y = 1|x, \beta_0, \beta_1) = \sigma(\beta_1 x + \beta_0)$



Example


Regression of Sex on Height
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-0.1
P
30 50 70
95
%Male
Regression Line
P=.50
70 50 30
5
%Female
40
50
60
70
80
90
Height in Inches

Parameter Estimation

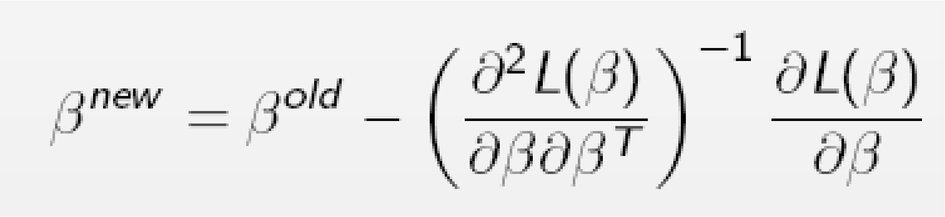
- MLE estimation
  - Given a dataset $D, with\ n\ data\ points$
  - For a single data object with attributes $\boldsymbol{x}_i$, class label y_i
    - Let $p(\boldsymbol{x}_i; \beta) = p_i = (Y = 1|\boldsymbol{x}_i, \beta), the\ prob.of\ \ i\ in\ class\ 1$
    - The probability of observing y_i would be
      - If $y_i = 1, then\ \ p_i$
      - If $y_i = 0, then\ 1 - p_i$
      - Combing the two cases: $p_i^{y_i}(1-p_i)^{1-y_i}$

$$L = \prod_i p_i^{y_i}(1-p_i)^{1-y_i} = \prod_i \left(\frac{\exp\{X^T\beta\}}{1+\exp\{X^T\beta\}}\right)^{y_i} \left(\frac{1}{1+\exp\{X^T\beta\}}\right)^{1-y_i}$$

Optimization

- Equivalent to maximize log likelihood
- $L = \sum_i y_i \boldsymbol{x}_i^T \beta - \log\left(1 + \exp\{\boldsymbol{x}_i^T \beta\}\right)$
- Gradient ascent update:
  - $$\beta^{new} = \beta^{old} + \boxed{\eta} \frac{\partial L(\beta)}{\partial \beta}$$

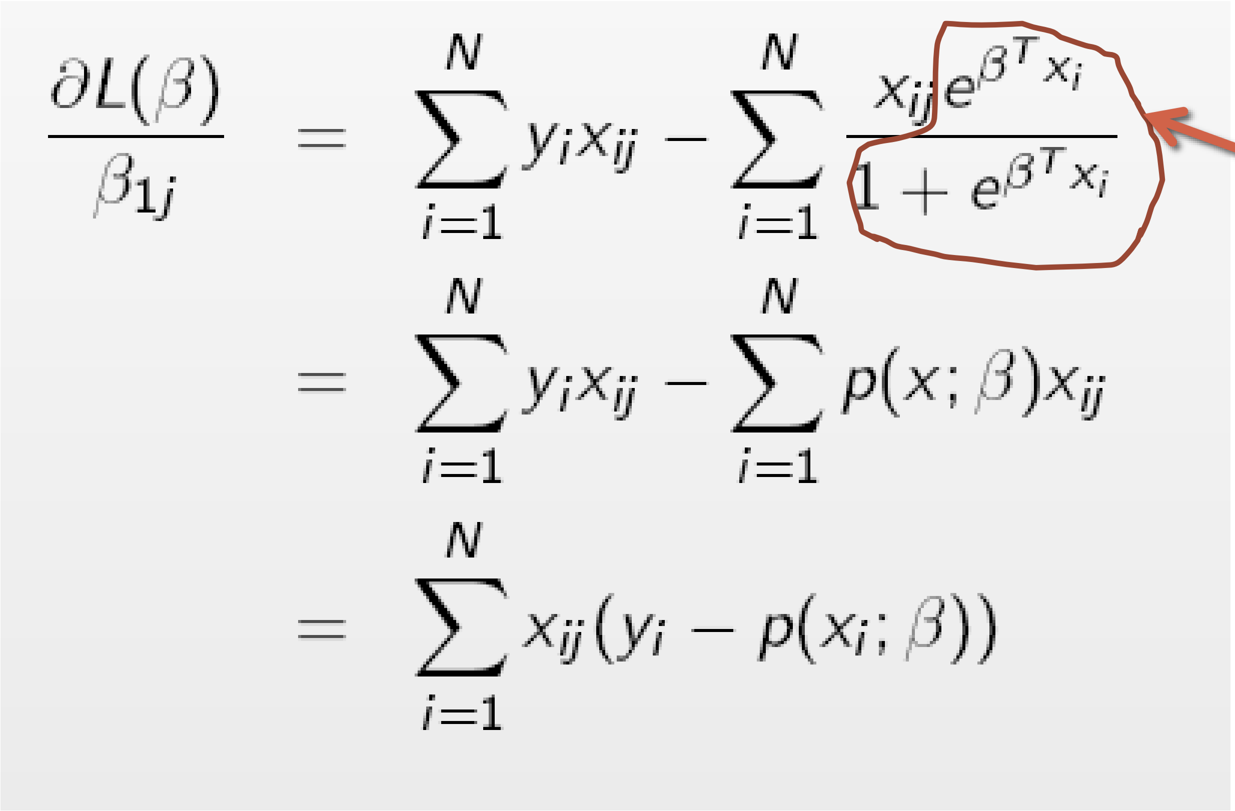
Step size, usually set as 0.1

- Newton-Raphson update
  - 

$$\beta^{new} = \beta^{old} - \left(\frac{\partial^2 L(\beta)}{\partial\beta\partial\beta^T}\right)^{-1} \frac{\partial L(\beta)}{\partial \beta}$$

  - where derivatives at evaluated at β^{old}

First Derivative



$$\frac{\partial L(\beta)}{\beta_{1j}} = \sum_{i=1}^{N} y_i x_{ij} - \sum_{i=1}^{N} \underbrace{\frac{x_{ij} e^{\beta^T x_i}}{1 + e^{\beta^T x_i}}}_{p(x_i; \beta)}$$

$$= \sum_{i=1}^{N} y_i x_{ij} - \sum_{i=1}^{N} p(x; \beta) x_{ij}$$

$$= \sum_{i=1}^{N} x_{ij} (y_i - p(x_i; \beta))$$

j = 0, 1, …, p

Second Derivative

- It is a (p+1) by (p+1) matrix, Hessian Matrix, with jth row and nth column as

$$
\begin{aligned}
& \frac{\partial L(\beta)}{\partial \beta_{1j} \partial \beta_{1n}} \\
= & -\sum_{i=1}^{N} \frac{(1+e^{\beta^T x_i})e^{\beta^T x_i} x_{ij} x_{in} - (e^{\beta^T x_i})^2 x_{ij} x_{in}}{(1+e^{\beta^T x_i})^2} \\
= & -\sum_{i=1}^{N} x_{ij} x_{in} p(x_i;\beta) - x_{ij} x_{in} p(x_i;\beta)^2 \\
= & -\sum_{i=1}^{N} x_{ij} x_{in} p(x_i;\beta)(1-p(x_i;\beta)) \,.
\end{aligned}
$$

What about Multiclass Classification?

- It is easy to handle under logistic regression, say M classes
  - $P(Y = j|X\ ) = \frac{\exp\{X^T\beta_j\}}{1+\sum_{m=1}^{M-1}\exp\{X^T\beta_m\}}$, for j $= 1, \ldots, M-1$
  - $P(Y = M|X) = \frac{1}{1+\sum_{m=1}^{M-1}\exp\{X^T\beta_m\}}$

Summary

- Bayesian Learning
  - Bayes theorem
  - Naïve Bayes, class conditional independence
  - Bayesian Belief Network, DAG, conditional probability table
- Logistic Regression
  - Logistic function, two-class logistic regression, MLE estimation, Gradient ascent updte, Newton-Raphson update, multiclass classification under logistic regression