

# CS6220: DATA MINING TECHNIQUES

## Matrix Data: Classification: Part 2

---


**Instructor: Yizhou Sun**

[yzsun@ccs.neu.edu](mailto:yzsun@ccs.neu.edu)

October 1, 2013

# Matrix Data: Classification: Part 2

---

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

# Bayesian Classification: Why?

---

- A statistical classifier: performs *probabilistic prediction*, i.e., predicts class membership probabilities
- Foundation: Based on Bayes' Theorem.
- Performance: A simple Bayesian classifier, *naïve Bayesian classifier*, has comparable performance with decision tree and selected neural network classifiers
- Incremental: Each training example can incrementally increase/decrease the probability that a hypothesis is correct — prior knowledge can be combined with observed data
- Standard: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

# Basic Probability Review

---

- Have two dices  $h_1$  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  $P(X,Y)$ , then the so-called marginal probability  $P(X)$  can be computed as 
$$P(X) = \sum_Y P(X,Y)$$
- For any events  $X$  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  $\mathbf{X}$  be a data sample (“*evidence*”)
- Let  $h$  be a *hypothesis* that  $\mathbf{X}$  belongs to class  $C$
- $P(h)$  (*prior probability*): the initial probability
  - E.g.,  $\mathbf{X}$  will buy computer, regardless of age, income, ...
- $P(\mathbf{X}|h)$  (likelihood): the probability of observing the sample  $\mathbf{X}$ , given that the hypothesis holds
  - E.g., Given that  $\mathbf{X}$  will buy computer, the prob. that  $\mathbf{X}$  is 31..40, medium income
- $P(\mathbf{X})$ : marginal probability that sample data is observed
  - $P(\mathbf{X}) = \sum_h P(\mathbf{X}|h) P(h)$
- $P(h|\mathbf{X})$ , (i.e., *posteriori probability*): the probability that the hypothesis holds given the observed data sample  $\mathbf{X}$

# Classification: Choosing Hypotheses

---

- *Maximum Likelihood* (maximize the likelihood):

$$h_{ML} = \arg \max_{h \in H} P(D | h)$$

- *Maximum a posteriori* (maximize the posterior):

- Useful observation: it does not depend on the denominator  $P(D)$

$$h_{MAP} = \arg \max_{h \in H} P(h | D) = \arg \max_{h \in H} P(D | h)P(h)$$

**$D$ : the whole training data set**

# 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  $\mathbf{X} = (x_1, x_2, \dots, x_p)$
- Suppose there are  $m$  classes  $Y \in \{C_1, C_2, \dots, C_m\}$
- Classification is to derive the maximum posteriori, i.e., the maximal  $P(Y=C_j | \mathbf{X})$
- 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  $P(\mathbf{X})$  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 (true positive);
  - a correct negative result in only 97% of the cases (true negative).
  - 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

---

$$P(\text{cancer}) = .008$$

$$P(\neg \text{cancer}) = .992$$

$$P(+ve | \text{cancer}) = .98 \quad P(-ve | \text{cancer}) = .02$$

$$P(+ve | \neg \text{cancer}) = .03 \quad P(-ve | \neg \text{cancer}) = .97$$

Using Bayes Formula:

$$P(\text{cancer} | +ve) = P(+ve | \text{cancer}) \times P(\text{cancer}) / P(+ve)$$

$$= 0.98 \times 0.008 / P(+ve) = .00784 / P(+ve)$$

$$P(\neg \text{cancer} | +ve) = P(+ve | \neg \text{cancer}) \times P(\neg \text{cancer}) / P(+ve)$$

$$= 0.03 \times 0.992 / P(+ve) = .0298 / P(+ve)$$

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  $\mathbf{X} = (x_1, x_2, \dots, x_p)$
- Suppose there are  $m$  classes  $Y \in \{C_1, C_2, \dots, C_m\}$
- Goal: Find  $Y$   
 $\max P(Y|\mathbf{X}) = P(Y, \mathbf{X})/P(\mathbf{X}) \propto P(\mathbf{X}|Y)P(Y)$
- A simplified assumption: attributes are **conditionally independent given the class** (class conditional independency):

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

# Estimate Parameters by MLE

---

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

$$\begin{aligned} L &= \prod_i P(\mathbf{X}_i, Y_i) = \prod_i P(\mathbf{X}_i | Y_i) P(Y_i) \\ &= \prod_i \left( \prod_k P(X_{ik} | Y_i) \right) P(Y_i) \end{aligned}$$

- 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  $\mu$  and standard deviation  $\sigma$
  - Estimate  $(\mu, \sigma^2)$  according to the observed  $X$  in the category of  $C_j$ 
    - Sample mean and sample variance
  - $P(x_k | C_j)$  is then

$$P(X_k = x_k | C_j) = g(x_k, \mu_{C_j}, \sigma_{C_j})$$

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	com
<=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

- $P(C_i)$ :  $P(\text{buys\_computer} = \text{"yes"}) = 9/14 = 0.643$

$$P(\text{buys\_computer} = \text{"no"}) = 5/14 = 0.357$$

- Compute  $P(X|C_i)$  for each class

$$P(\text{age} = \text{"<=30"} \mid \text{buys\_computer} = \text{"yes"}) = 2/9 = 0.222$$

$$P(\text{age} = \text{"<= 30"} \mid \text{buys\_computer} = \text{"no"}) = 3/5 = 0.6$$

$$P(\text{income} = \text{"medium"} \mid \text{buys\_computer} = \text{"yes"}) = 4/9 = 0.444$$

$$P(\text{income} = \text{"medium"} \mid \text{buys\_computer} = \text{"no"}) = 2/5 = 0.4$$

$$P(\text{student} = \text{"yes"} \mid \text{buys\_computer} = \text{"yes"}) = 6/9 = 0.667$$

$$P(\text{student} = \text{"yes"} \mid \text{buys\_computer} = \text{"no"}) = 1/5 = 0.2$$

$$P(\text{credit\_rating} = \text{"fair"} \mid \text{buys\_computer} = \text{"yes"}) = 6/9 = 0.667$$

$$P(\text{credit\_rating} = \text{"fair"} \mid \text{buys\_computer} = \text{"no"}) = 2/5 = 0.4$$

- **$X = (\text{age} \leq 30, \text{income} = \text{medium}, \text{student} = \text{yes}, \text{credit\_rating} = \text{fair})$**

$$P(X|C_i) : P(X \mid \text{buys\_computer} = \text{"yes"}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$$

$$P(X \mid \text{buys\_computer} = \text{"no"}) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$$

$$P(X|C_i) * P(C_i) : P(X \mid \text{buys\_computer} = \text{"yes"}) * P(\text{buys\_computer} = \text{"yes"}) = 0.028$$

$$P(X \mid \text{buys\_computer} = \text{"no"}) * P(\text{buys\_computer} = \text{"no"}) = 0.007$$

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  $x_k = v$ ,  $V$  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:  $X_k | C_j \sim \theta$* 
  - $P(X_k = v | C_j, \theta) = \theta_v$
- Put prior to  $\theta$ 
  - In discrete case, the prior can be chosen as symmetric Dirichlet distribution:  $\theta \sim \text{Dir}(\alpha)$ , i. e.,  $P(\theta) \propto \prod_v \theta_v^{\alpha-1}$
  - *posterior distribution:*
    - $P(\theta | X_{1k}, \dots, x_{nk}, C_j) \propto P(X_{1k}, \dots, x_{nk} | C_j, \theta) P(\theta)$ , **another Dirichlet distribution**, with new parameter  $(\alpha + c_1, \dots, \alpha + c_v, \dots, \alpha + c_V)$
    - $c_v$  is the number of observations taking value  $v$
  - Inference:
$$P(X_k = v | X_{1k}, \dots, x_{nk}, C_j) = \int P(X_k = v | \theta) P(\theta | X_{1k}, \dots, x_{nk}, C_j) d\theta$$
$$= \frac{c_v + \alpha}{\sum c_v + V\alpha}$$
    - Equivalent to adding  $\alpha$  to each observation value  $v$

# \*Notes on Parameter Learning

---

- Why the probability of  $P(X_k | C_i)$  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

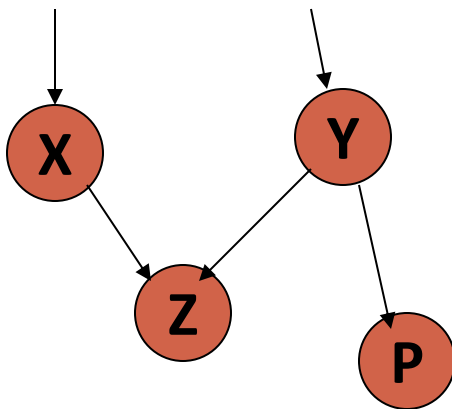
# Matrix Data: Classification: Part 2

---

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

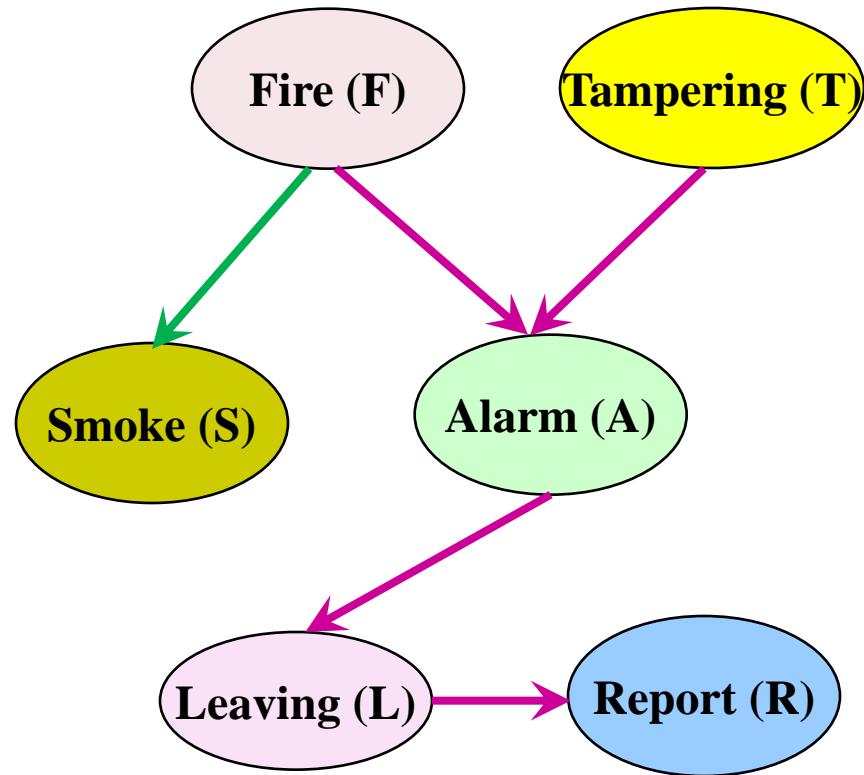
# Bayesian Belief Networks (BNs)

- **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

# 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  $\mathbf{X}$ , from CPT (**joint probability**):

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{Parents}(x_i))$$

# Inference in Bayesian Networks

---

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

# Inference by enumeration

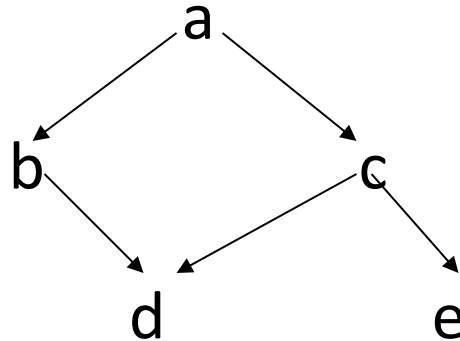
---

- To compute posterior marginal  $P(X_i | E=e)$ 
  - Add all of the terms (atomic event probabilities) from the full joint distribution
  - If  $\mathbf{E}$  are the evidence (observed) variables and  $\mathbf{Y}$  are the other (unobserved) variables, then:  
$$P(X|\mathbf{e}) = \alpha P(X, \mathbf{E}) = \alpha \sum P(X, \mathbf{E}, \mathbf{Y})$$
  - Each  $P(X, \mathbf{E}, \mathbf{Y})$  term can be computed using the chain rule
- Computationally expensive!



# Example: Enumeration

---



- $P(d|e) = \alpha \sum_{ABC} P(a, b, c, d, e)$   
 $= \alpha \sum_{ABC} P(a) P(b|a) P(c|a) P(d|b,c) P(e|c)$
- 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.,  $S \leftarrow F \rightarrow A \leftarrow T$ , path  $S \rightarrow A$  is blocked once we know  $F \rightarrow A$
- **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

- *Y: continuous value  $(-\infty, +\infty)$*

- $Y = \mathbf{x}^T \boldsymbol{\beta} = \beta_0 + x_1\beta_1 + x_2\beta_2 + \dots + x_p\beta_p$

- $Y|\mathbf{x}, \boldsymbol{\beta} \sim N(\mathbf{x}^T \boldsymbol{\beta}, \sigma^2)$

- Logistic Regression

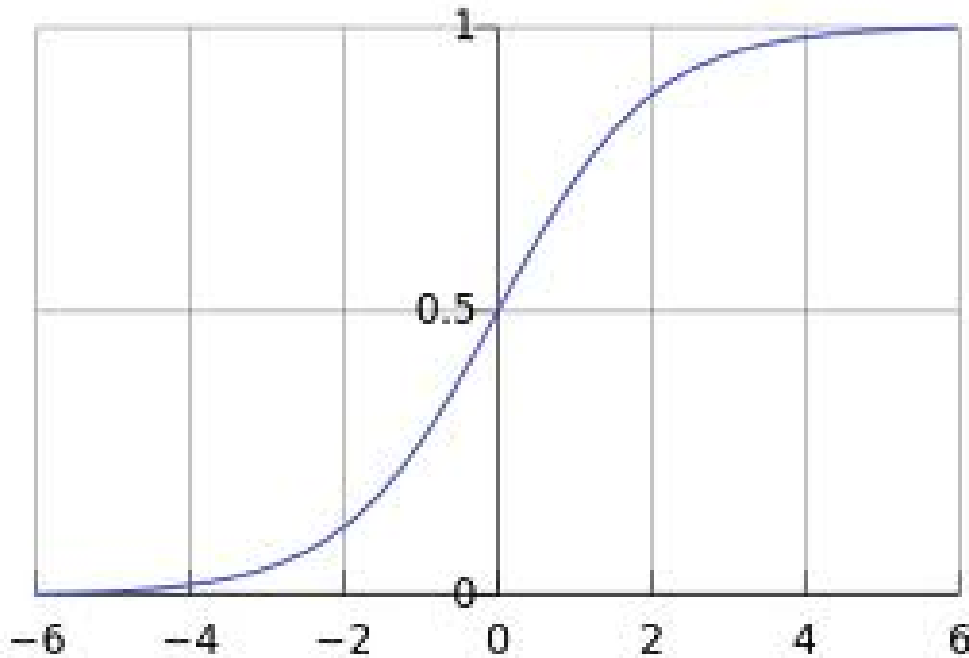
- *Y: discrete value from  $m$  classes*

- $p(Y = C_j) \in (0,1)$  and  $\sum_j p(Y = C_j) = 1$

# Logistic Function

---

- Logistic Function:  $f(x) = \frac{1}{1+e^{-x}}$
- A special case of sigmoid function



# Modeling Probabilities of Two Classes

---

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

$$\beta = \begin{pmatrix} \beta_{10} \\ \beta_{11} \\ \beta_{12} \\ \vdots \\ \beta_{1,p} \end{pmatrix}$$

- In other words

- $Y|X, \beta \sim \text{Bernoulli}\left(\frac{1}{1 + \exp\{-X^T \beta\}}\right)$

# Parameter Estimation

---

- MLE estimation
  - Given a dataset  $D$ , with  $n$  data points
  - For a single data object with attributes  $\mathbf{x}_i$ , class label  $y_i$ 
    - Let  $p(\mathbf{x}_i; \beta) = p_i = P(Y = 1 | \mathbf{x}_i, \beta)$ , the prob. of 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
- $L = \sum_i y_i \mathbf{x}_i^T \beta - \log(1 + \exp\{\mathbf{x}_i^T \beta\})$
- 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 are evaluated at  $\beta^{old}$

# First Derivative

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

$$j = 0, 1, \dots, p$$

# Second Derivative

- It is a  $(p+1)$  by  $(p+1)$  matrix, Hessian Matrix, with  $j$ th row and  $n$ th 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

- $$P(Y = j|X) = \frac{\exp\{X^T \beta_j\}}{1 + \sum_{m=1}^{M-1} \exp\{X^T \beta_m\}}, \text{ for } j = 1, \dots, M - 1$$

- $$P(Y = M|X) = \frac{1}{1 + \sum_{m=1}^{M-1} \exp\{X^T \beta_m\}}$$

# 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, Newton-Raphson update, multiclass classification under logistic regression