UVA CS 6316: Machine Learning

Lecture 14: Logistic Regression

Dr. Yanjun Qi

University of Virginia

Department of Computer Science

Course Content Plan ➔ Six major sections of this course

- ❑ ~~Regression (supervised)~~ — Y is a continuous
- ❑ Classification (supervised) — Y is a discrete
- ❑ Unsupervised models — NO Y
- ❑ Learning theory — About f()
- ❑ Graphical models — About interactions among X1,… Xp
- ❑ Reinforcement Learning — Learn program to Interact with its environment

Today

- ❑ Bayes Classifier
- ❑ Logistic Regression
- ❑ Training LG by MLE

Bayes Classifier

$c^* = \arg\max P(C_j \mid X_1, \dots X_p)$

Task
↓
Representation
↓
Score Function
↓
Search/Optimization
↓
Models, Parameters

Classification

$\{C_1, C_2, \dots, C_L\}$

↓

$$c_{MAP} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j \mid x_1, x_2, \dots, x_p)$$

↓

MLE

↓

?

↓

Parameters in P(C|X)

Bayes classifiers

- Treat each feature attribute and the class label as random variables.

$\{c_1, \dots, c_L\}$

Bayes classifiers

- Treat each feature attribute and the class label as random variables.

- Testing: Given a sample **x** with attributes ($x_1, x_2, \ldots, x_p$):
  - Goal is to predict its class *c*.
  - Specifically, we want to find the class that maximizes $p(c \mid x_1, x_2, \ldots, x_p)$.

- Training: can we estimate $p(C_i \mid \mathbf{x}) = p(C_i \mid x_1, x_2, \ldots, x_p)$ directly from data?

Bayes Classifiers – MAP Rule

Task: Classify a new instance *X* based on a tuple of attribute values $X = \langle X_1, X_2, \ldots, X_p \rangle$ into one of the classes

$$c_{MAP} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j \mid x_1, x_2, \ldots, x_p)$$

MAP Rule

MAP = Maximum Aposteriori Probability

Adapt From Carols' prob tutorial

Bayes Classifiers – MAP Classification Rule

- Establishing a probabilistic model for classification

➔ **MAP** classification rule

- **MAP**: **M**aximum **A** **P**osterior
- Assign x to c^* if

$$\sum_{j=1}^{L} P(C=c_j|x)=1$$

$$P(C=c^*|\mathbf{X}=\mathbf{x}) > P(C=c|\mathbf{X}=\mathbf{x})$$

$$\text{for } c \neq c^*, \; c=c_1,\cdots,c_L$$

Adapt from Prof. Ke Chen NB slides

$$f : X \longrightarrow C$$

Output as Discrete Class Label

$C_1, C_2, \ldots, C_L$

Establishing a probabilistic model for classification

$$c_{MAP} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j \mid x_1, x_2, \ldots, x_p)$$

$$\frac{P(X,c)}{P(X)}$$

Generative

$$\underset{c \in C}{\operatorname{argmax}} P(c \mid X) = \underset{c \in C}{\operatorname{argmax}} P(X, c) = \underset{c \in C}{\operatorname{argmax}} P(X \mid c) P(c)$$

Later!

Discriminative

$$\underset{c \in C}{\arg\max} P(c / \mathbf{X}) \quad C = \{c_1, \cdots, c_L\}$$

Recap: Statistical Decision Theory (Extra)

- Random input vector: X
- Random output variable: Y
- Joint distribution: Pr(X,Y) $\Rightarrow D = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\}$
- Loss function L(Y, f(X))

- Expected prediction error (EPE):

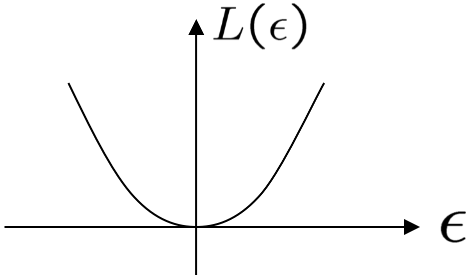
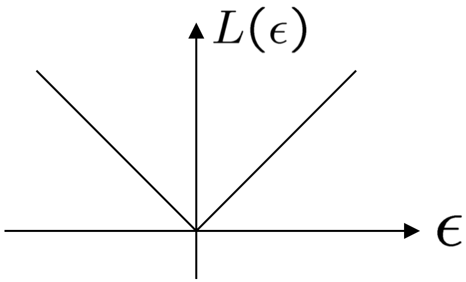
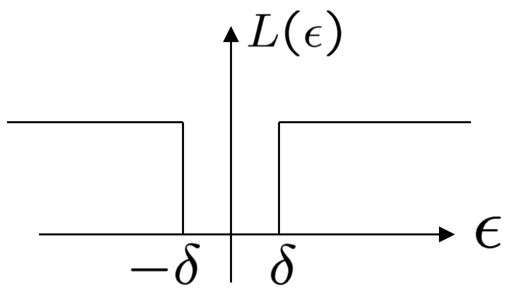
$$\mathrm{EPE}(f) = \mathrm{E}(L(Y, f(X))) = \int L(y, f(x)) \Pr(dx, dy)$$

$$\text{e.g.} = \int (y - f(x))^2 \Pr(dx, dy)$$

e.g. Squared error loss (also called L2 loss)

Consider population distribution

SUMMARY: WHEN Expected prediction error (EPE) USES DIFFERENT LOSS

Loss Function		Estimator $\hat{f}(x)$
L_2		$EPE = E_{X,Y}(Y - f(X))^2$ $\hat{f}(x) = E[Y \mid X = x]$
L_1		$\hat{f}(x) = \text{median}(Y \mid X = x)$
0-1		$\hat{f}(x) = \arg\max_Y P(Y \mid X = x)$ (Bayes classifier / MAP)

Please read extra slides for WHY MAP-rule makes sense

$$EPE(f) = E_{X,C}\left(L(C, f(X))\right)$$

$$E_C(C) = \sum_{i=1}^{L} C_i P(C_i)$$

$$= E_X E_{C|X}\left[L(C, f(X)) \mid X\right]$$

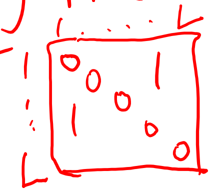
Discrete RV's Expectation

$$= E_X \sum_{k=1}^{L} L[C_k, f(X)] \Pr(C_k | X)$$

$$\underset{f}{\operatorname{argmin}}\ EPE(f(X))$$

$\Rightarrow$ Pointwise minization when $X = x$

$$\Rightarrow \hat{f}(X = x) = \underset{f(x) \in C}{\operatorname{argmin}} \sum_{k=1}^{L} L(C_k, f(x)) \Pr(C_k | X = x)$$



$$\Rightarrow \hat{f}(x) = \underset{C_k \in \{C_1, C_2, C_3, \ldots, C_L\}}{\operatorname{argmax}} \Pr(C_k | X = x)$$

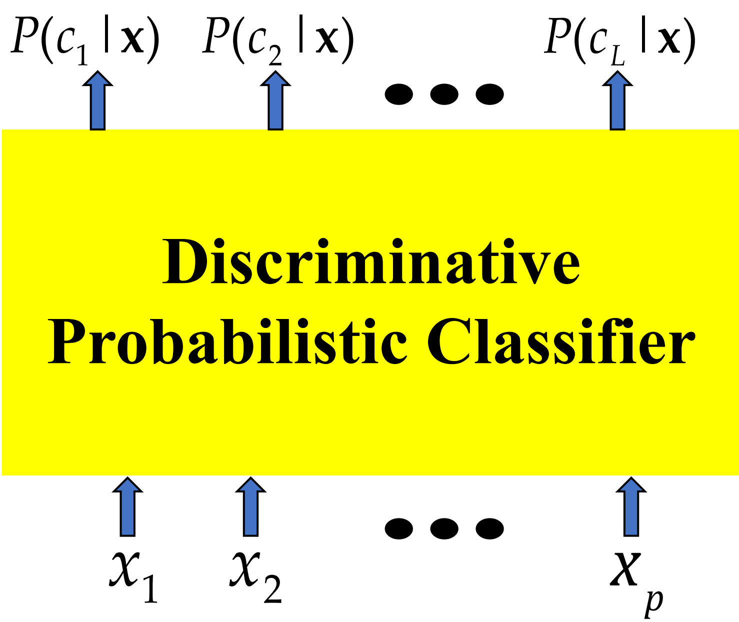
$$\left\{\begin{array}{l} p(C_1 | x) \\ p(C_2 | x) \\ \vdots \\ p(C_L | x) \end{array}\right.$$

Today:

$X \rightarrow C$: $p(c|x)$ → $f(x)$

– **Discriminative model**

$$\arg\max_{c \in C} P(c/\mathbf{X}), \quad C = \{c_1, \cdots, c_L\}$$



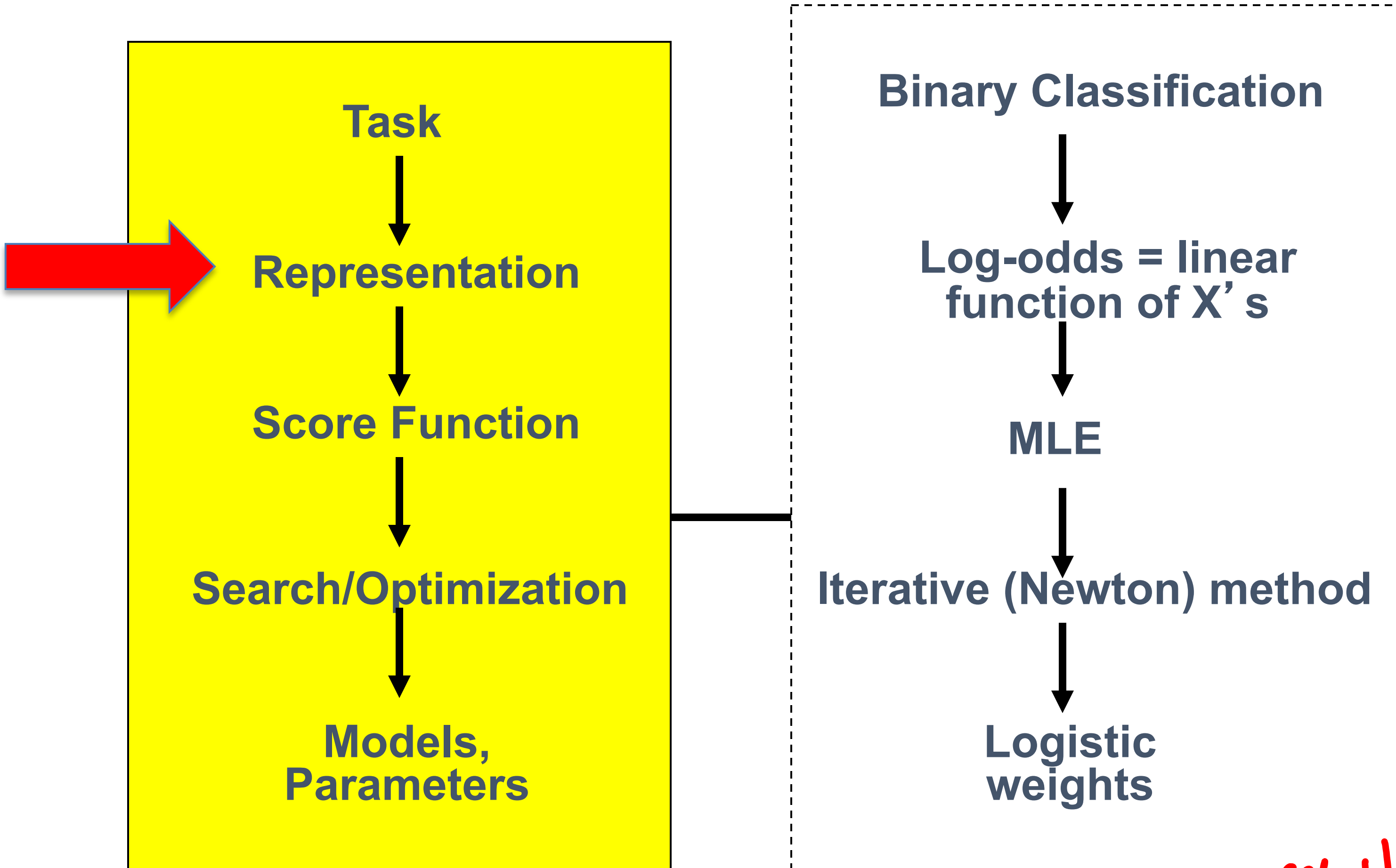
$$\mathbf{x} = (x_1, x_2, \cdots, x_p)$$

Adapt from Prof. Ke Chen NB slides

Today

- ❑ Bayes Classifier
- ❑ Logistic Regression
- ❑ Training LG by MLE

Logistic Regression



$$P(y=0|x) = 1 - P(y=1|x)$$

$$P(y=1|x) = \frac{e^{\beta_0 + \beta^T x}}{1 + e^{\beta_0 + \beta^T x}}$$

Multivariate linear regression to Logistic Regression

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

Logistic regression for binary classification

$$\ln\left[\frac{P(y|x)}{1-P(y|x)}\right] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

=1

$$\log\left(\frac{P(y=1|x)}{P(y=0|x)}\right)$$

Logistic Regression p(y|x)

$$\ln\left[\frac{P(y|x)}{1-P(y|x)}\right]=\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_p x_p$$

$$P(y|x)=\frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_p x_p}}{1+e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_p x_p}}=\frac{1}{1+e^{-(\beta_0+\beta^T X)}}$$

The logit function View (e.g. when with 1D x)

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

logistic

$$\ln\left[\frac{P(y|x)}{1-P(y|x)}\right] = \alpha + \beta x$$

$\ln\left(\frac{p}{1-p}\right)$, $p = P(y=1|x)$

logit / log-odd

Logit function

Logit of $P(y|x)$

Binary Logistic Regression (Two Views)

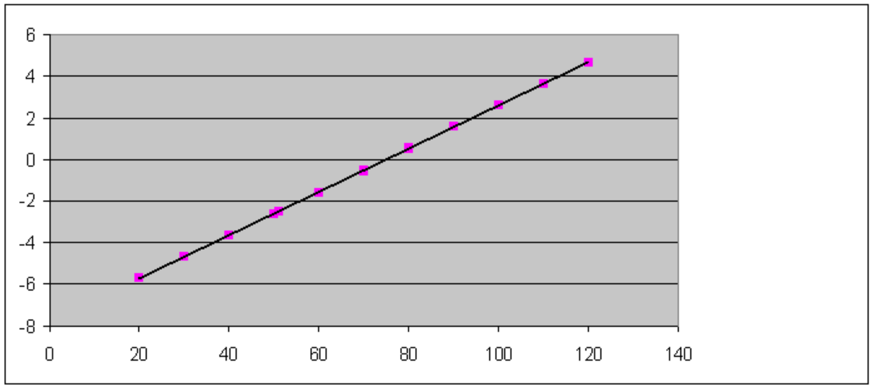
Bernoulli Distribution

p_{Head}

⇓

$Y \in \{0, 1\}$ (T, H)

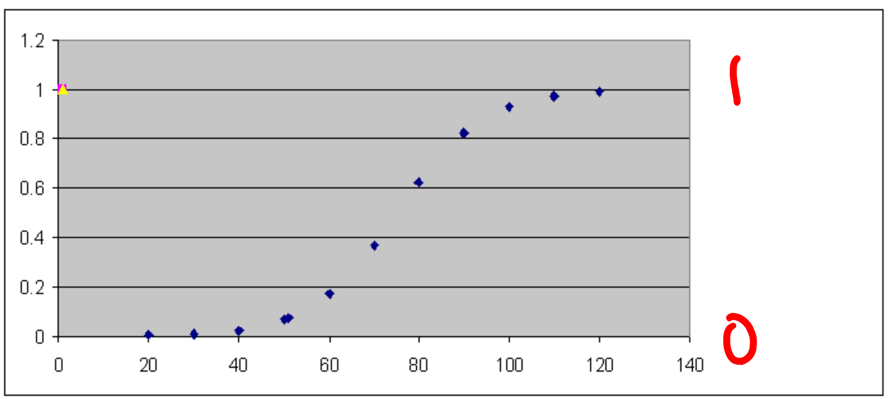
ln[p/(1-p)]



x

$p_{Head} = p(y=1|x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$

P (Y=1|x)



1

0

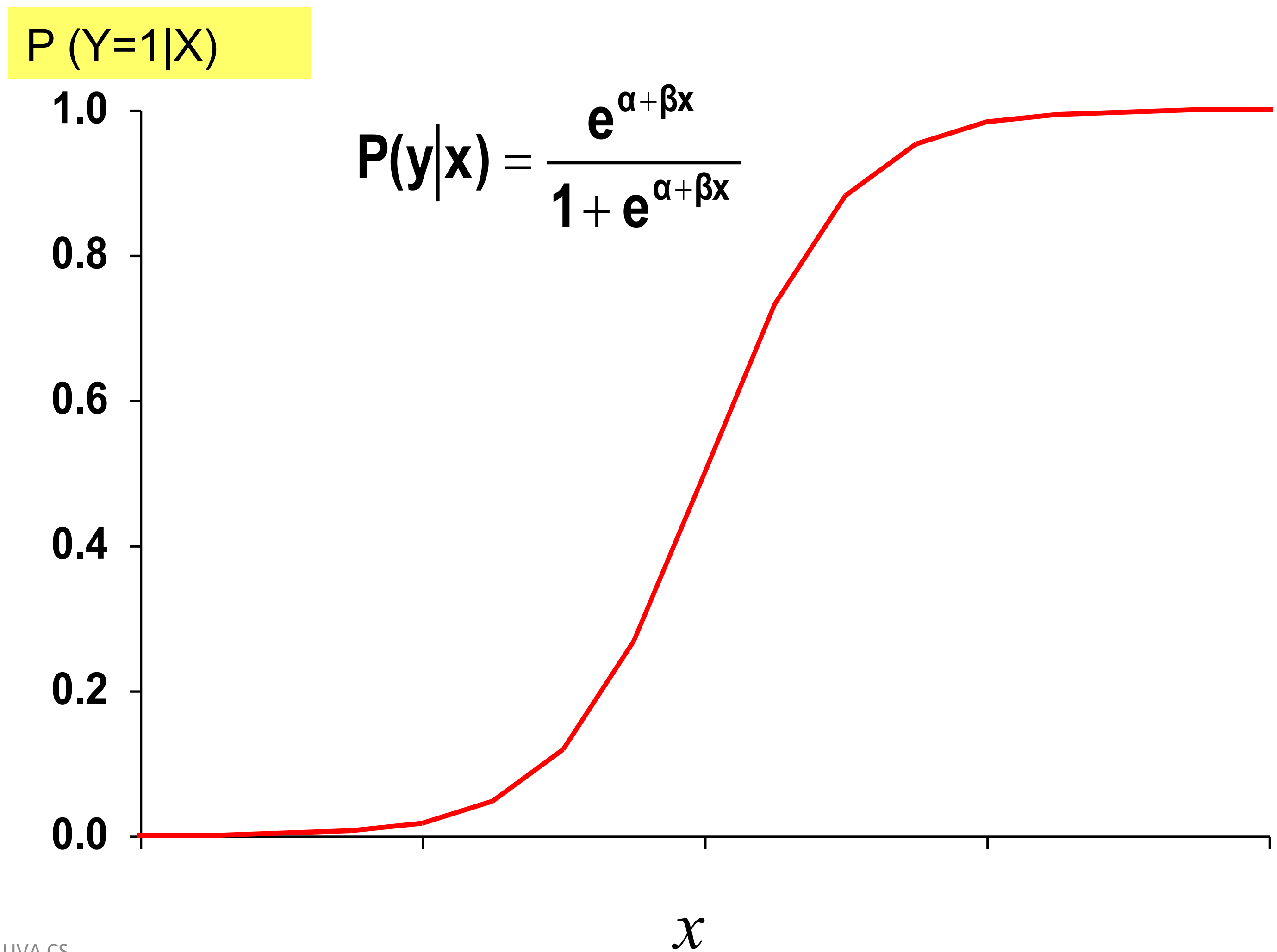
shape

x

P(y=1|x)   1-p(y=1x)

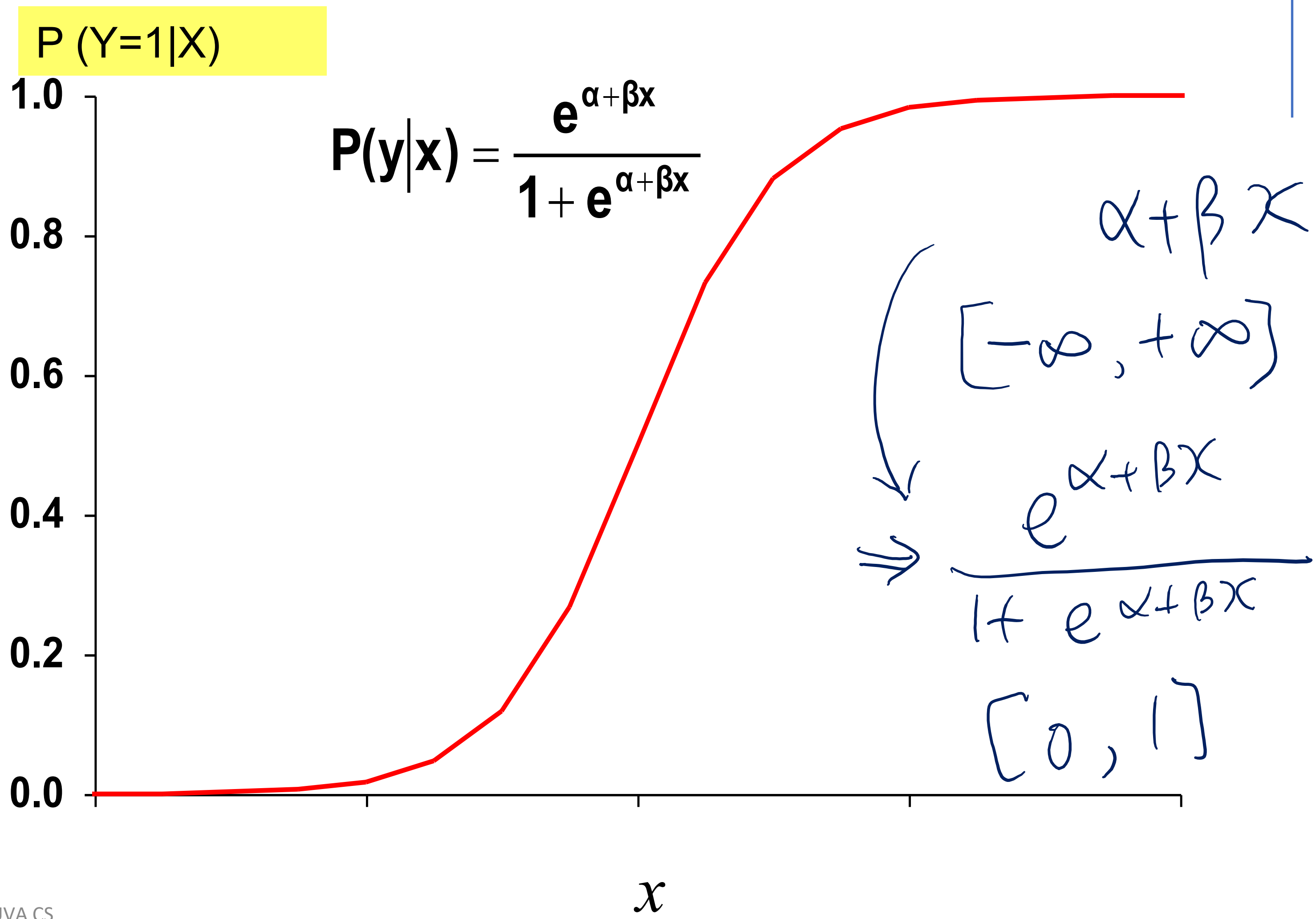
View I: logit of p(y=1|x) is linear function of x

e.g.
Probability of
disease

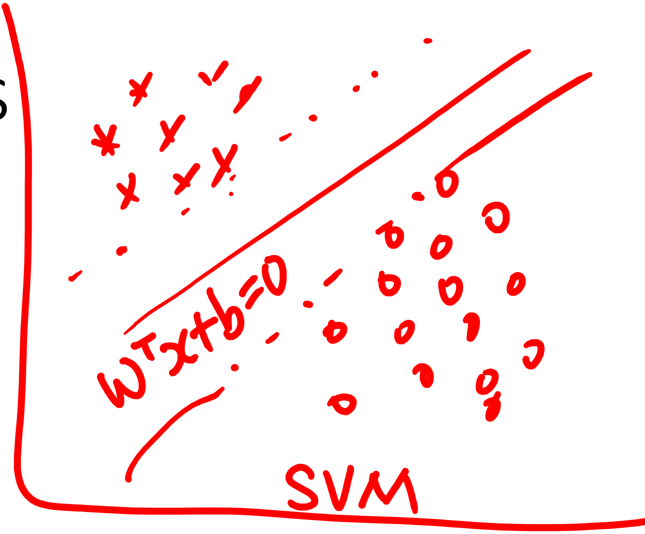


View II: "S" shape function compress output to [0,1]

e.g.
Probability of
disease



View III: Logistic Regression models a linear classification boundary!



$y = \{H, T\}$
$\quad\;\; \{1, 0\}$

$\underset{y \in \{0,1\}}{\operatorname{argmax}} \; p(y \mid \underline{x})$

$p(y=0 \mid x) = p(y=1 \mid x) \Rightarrow$ Decision Boundary $\dfrac{p(y=1 \mid x)}{p(y=0 \mid x)} = 1$

$\log\left(\dfrac{p(y=1 \mid x)}{p(y=0 \mid x)}\right) = \vec{\beta}^T \vec{x} = \log(1) = 0$

Logistic Regression models a linear classification boundary!

$$y \in \{0,1\}$$

$$\ln\left[\frac{P(y|x)}{1-P(y|x)}\right] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

Decision Boundary → equals to zero

$$\ln\left[\frac{P(y=1|x)}{P(y=0|x)}\right] = \ln\left[\frac{P(y=1|x)}{1-P(y=1|x)}\right] = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

Logistic Regression models a linear classification boundary!

[Seperate two classes]

$$\ln \frac{P(y=1|x)}{1-P(y=1|x)} = \ln \frac{P(y=1|x)}{P(y=0|x)} = 0$$

0.5

0.5

linear hyperplane

$$\alpha + \beta_1 x_1 + \cdots + \beta_p x_p = 0$$

↑

Boundary points

$$p(y=1|x) = p(y=0|x)$$

Logistic Regression—when?

⇒ y is model with Bernoulli (p)

Logistic regression models are appropriate when the target variable is coded as 0/1.

⇒ p is a func of x

We only observe "0" and "1" for the target variable—but we think of the target variable conceptually as a probability that "1" will occur.

This means we use Bernoulli distribution to model the target variable with its Bernoulli parameter p=p(y=1|x) predefined.

The main interest ➔ predicting the probability that an event occurs (i.e., the probability that p(y=1|x)).

Logistic Regression Assumptions

- Linearity in the logit – the regression equation should have a linear relationship with the logit form of the target variable

- There is no assumption about the feature variables / target predictors being linearly related to each other.

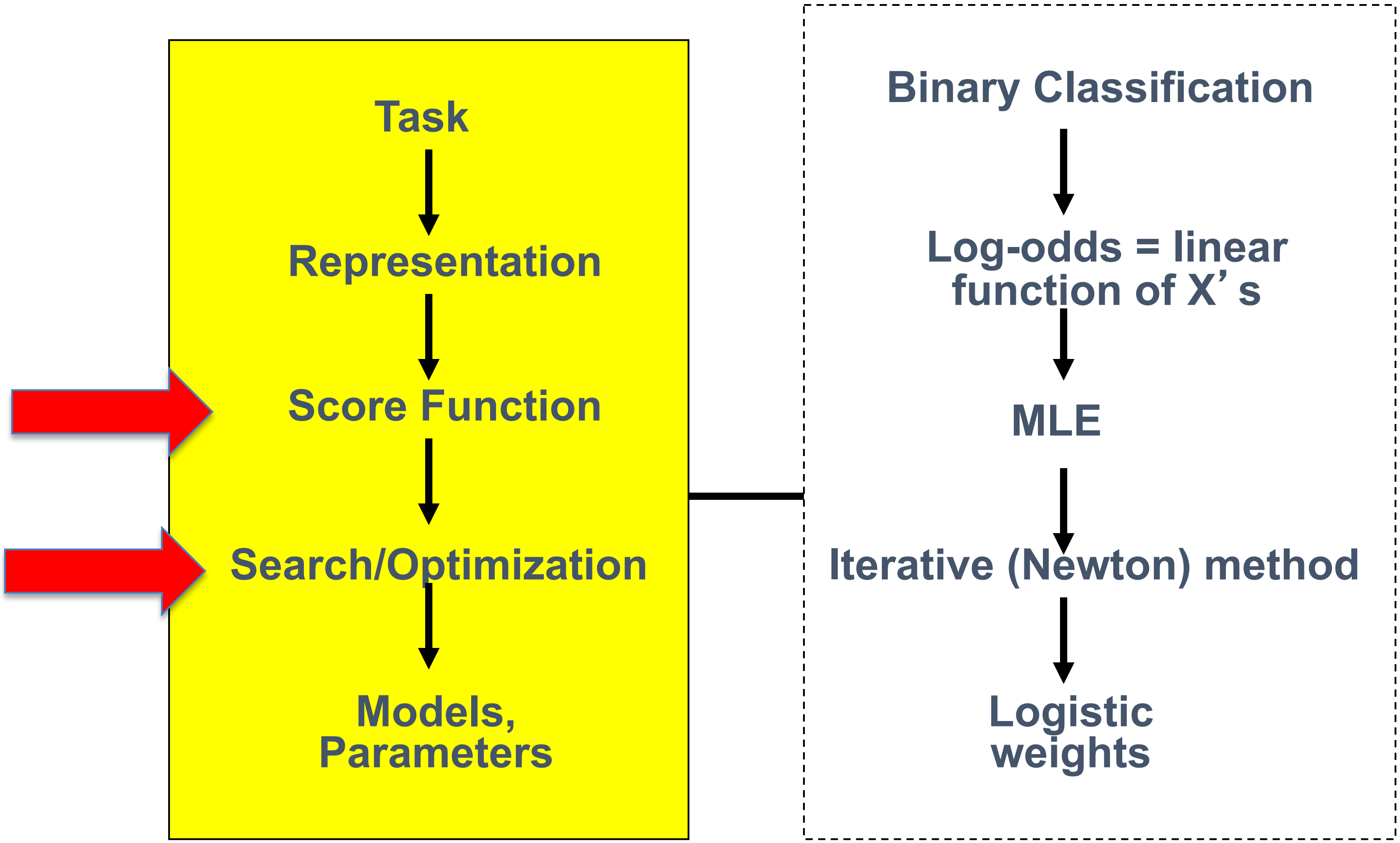
$P(y=1|x)$ $1-p(y=1x)$

func of x
with parameter $\vec{\beta}$ to learn from training data

Today

- ❑ Bayes Classifier
- ❑ Logistic Regression
- ❑ Training LG by MLE

Logistic Regression



$$P(y=1|x) = \frac{e^{\beta_0+\beta^T x}}{1+e^{\beta_0+\beta^T x}}$$

Review: Maximum Likelihood Estimation

A general Statement

Consider a sample set $T=(Z_1 \ldots Z_n)$ which is drawn from a probability distribution P(Z|\theta) where \theta are parameters.

If the Zs are independent with probability density function $P(Z_i|$\theta), the joint probability of the whole set is

$$P(Z_1 \ldots Z_n|\theta) = \prod_{i=1}^{n} P(Z_i|\theta)$$

this may be maximised with respect to \theta to give the maximum likelihood estimates.

The idea is to

✓ assume a particular model with unknown parameters, θ

The idea is to

- ✓ assume a particular model with unknown parameters, θ
- ✓ we can then define the probability of observing a given event conditional on a particular set of parameters. $P(Z_i|\theta)$

The idea is to

- ✓ assume a particular model with unknown parameters, θ
- ✓ we can then define the probability of observing a given event conditional on a particular set of parameters. $P(Z_i|\theta)$
- ✓ We have observed a set of outcomes in the real world.

The idea is to

- ✓ assume a particular model with unknown parameters, θ
- ✓ we can then define the probability of observing a given event conditional on a particular set of parameters. $P(Z_i|\theta)$
- ✓ We have observed a set of outcomes in the real world. $z_1, z_2, \ldots, z_n$
- ✓ It is then possible to choose a set of parameters which are most likely to have produced the observed results.

The idea is to

- ✓ assume a particular model with unknown parameters, θ
- ✓ we can then define the probability of observing a given event conditional on a particular set of parameters. $P(Z_i|\theta)$
- ✓ We have observed a set of outcomes in the real world.
- ✓ It is then possible to choose a set of parameters which are most likely to have produced the observed results.

$$\hat{\theta} = \underset{\theta}{argmax} \; P(Z_1 \ldots Z_n|\theta) = \prod_{i=1}^{n} P(Z_i|\theta)$$

This is maximum likelihood. In most cases it is both consistent and efficient.

$$log(L(\theta)) = \sum_{i=1}^{n} log(P(Z_i|\theta))$$

It is often convenient to work with the Log of the likelihood function.

The idea is to

- ✓ assume a particular model with unknown parameters, θ
- ✓ we can then define the probability of observing a given event conditional on a particular set of parameters. $P(Z_i|\theta)$
- ✓ We have observed a set of outcomes in the real world.
- ✓ It is then possible to choose a set of parameters which are most likely to have produced the observed results.

$$\hat{\theta} = \underset{\theta}{argmax} \; P(Z_1 \ldots Z_n|\theta)$$

Likelihood

This is maximum likelihood. In most cases it is both consistent and efficient.

$$log(L(\theta)) = \sum_{i=1}^{n} log(P(Z_i|\theta))$$

Log-Likelihood

It is often convenient to work with the Log of the likelihood function.

Review: Defining Likelihood for basic Bernoulli

Given: $\{z_1, z_2, \ldots, z_n\}$

$\Downarrow$

$\{H, H, T, \ldots, H\}_n$

$\Downarrow$ reformulate

$\{1, 1, 0, \ldots, 1\}_n$

constant $\theta = \{p\} = \{p(\text{Head})\}$

$p(z_i \mid \underline{\underline{\theta}}) = p^{z_i}(1-p)^{1-z_i}$ (Here $z_i \in \{0, 1\}$)

$$p(z_i) = \begin{cases} p, & \text{if } z_i = H/1 \\ 1-p, & \text{if } z_i = T/0 \end{cases} \Rightarrow \underset{p}{\operatorname{argmax}} \prod_{i=1}^{n} p^{z_i}(1-p)^{1-z_i}$$

Defining Likelihood

{H, H, T, . . . H}$_n$

Logistic Regression $z_i = y_i | x_i$

$\{y_1|x_1, y_2|x_2, \ldots, y_n|x_n\}$

Observing binary samples z_i

PMF:

$p(z_i|\beta)$

$= p(y_i|x_i, \beta)$

$$\Pr(z_i|p) = p^{z_i}(1-p)^{1-z_i}$$

Now we just rewrite

$\hat{y}_i = p(y=1|x_i)$

LIKELIHOOD:

$$L(p) = \prod_{i=1}^{n} p^{z_i}(1-p)^{1-z_i}$$

$\Rightarrow p(z_i|\beta) = \hat{y}_i^{\,y_i}(1-\hat{y}_i)^{1-y_i}$

function of p=Pr(head)

Basic Bernoulli | Logistic / Bernoulli

LIKELIHOOD:

$$L(p) = \prod_{i=1}^{n} p^{z_i} (1-p)^{1-z_i}$$

function of p=Pr(head)

$$L(\beta) = \prod_{i=1}^{n} p(y_i=1|x_i)^{y_i} (1-p(y_i=1|x_i))^{1-y_i}$$

$$= \prod_{i=1}^{n} \hat{y}_i^{\,y_i} (1-\hat{y}_i)^{1-y_i}$$

$$\log(L(p)) = \log\left[\prod_{i=1}^{n} p^{z_i} (1-p)^{1-z_i}\right]$$

$$= \sum_{i=1}^{n} (z_i \log p + (1-z_i)\log(1-p))$$

Loglikelihood

$$\ell\ell(\beta) = \sum_{i=1}^{n} \left(y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i) \right)$$

$$\ell\ell\, l(\beta) = \sum_{i=1}^{N} \{\log \Pr(Y = y_i \mid X = x_i)\}$$

When training set includes (x_i, y_i), i=1,...,N

$$\ell\ell(\beta) = \sum_{i=1}^{N} \log P(y_i \mid x_i)$$

Here $P(y_i \mid x_i) = \begin{cases} p(y=1 \mid x_i), & \text{if } y_i = 1 \\ p(y=0 \mid x_i), & \text{if } y_i = 0 \end{cases}$

$$= \left(p(y=1 \mid x_i)\right)^{y_i} \left(1 - p(y=1 \mid x_i)\right)^{1-y_i}$$

MLE for Logistic Regression Training

Training set: (x_i, y_i), i=1,…,N

$$l(\beta) = \sum_{i=1}^{N} \{\log \Pr(Y = y_i \mid X = x_i)\}$$

$$= \sum_{i=1}^{N} \{y_i \log(\Pr(Y=1 \mid X = x_i)) + (1-y_i)\log(\Pr(Y=0 \mid X=x_i))\}$$

$$= \sum_{i=1}^{N} (y_i \log \frac{\exp(\beta^T x_i)}{1+\exp(\beta^T x_i)}) + (1-y_i)\log \frac{1}{1+\exp(\beta^T x_i)})$$

$$= \sum_{i=1}^{N} (y_i \beta^T x_i - \log(1+\exp(\beta^T x_i)))$$

Cross entropy loss $\sum_{i=1}^{n} y_i \log \hat{y}_i + (1-y_i) \log (1-\hat{y}_i)$

Summary: MLE for Logistic Regression Training

Let's fit the logistic regression model for *K*=2, i.e., number of classes is 2

Training set: (x_i, y_i), i=1,…,*N*

(conditional) Log-likelihood:

How?

For Bernoulli distribution

$$p(y|x)^y(1-p)^{1-y}$$

$$l(\beta) = \sum_{i=1}^{N} \{\log \Pr(Y = y_i | X = x_i)\}$$

$$= \sum_{i=1}^{N} y_i \log(\Pr(Y=1|X=x_i)) + (1-y_i)\log(\Pr(Y=0|X=x_i))$$

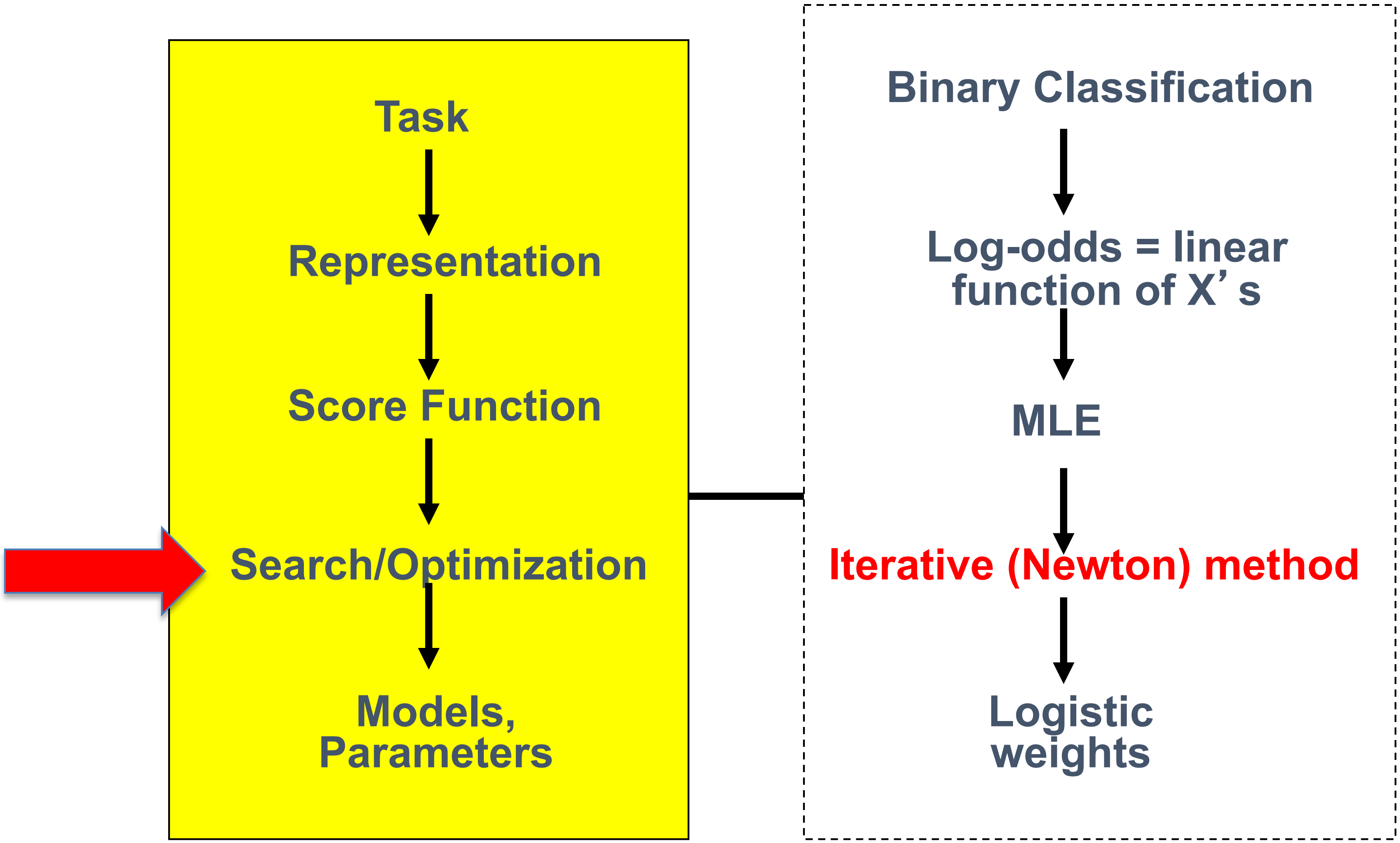
$$= \sum_{i=1}^{N} (y_i \log \frac{\exp(\beta^T x_i)}{1+\exp(\beta^T x_i)}) + (1-y_i)\log \frac{1}{1+\exp(\beta^T x_i)})$$

$$= \sum_{i=1}^{N} (y_i \beta^T x_i - \log(1+\exp(\beta^T x_i)))$$

x_i are (*p*+1)-dimensional input vector with leading entry 1
\beta is a (*p*+1)-dimensional vector

We want to maximize the log-likelihood in order to estimate *beta*

Logistic Regression



$$P(y=1|x) = \frac{e^{\beta_0+\beta^T x}}{1+e^{\beta_0+\beta^T x}}$$

MLE for Logistic Regression Training

Training set: (x_i, y_i), i=1,…,N

$$l(\beta) = \sum_{i=1}^{N} \{\log \Pr(Y = y_i \mid X = x_i)\}$$

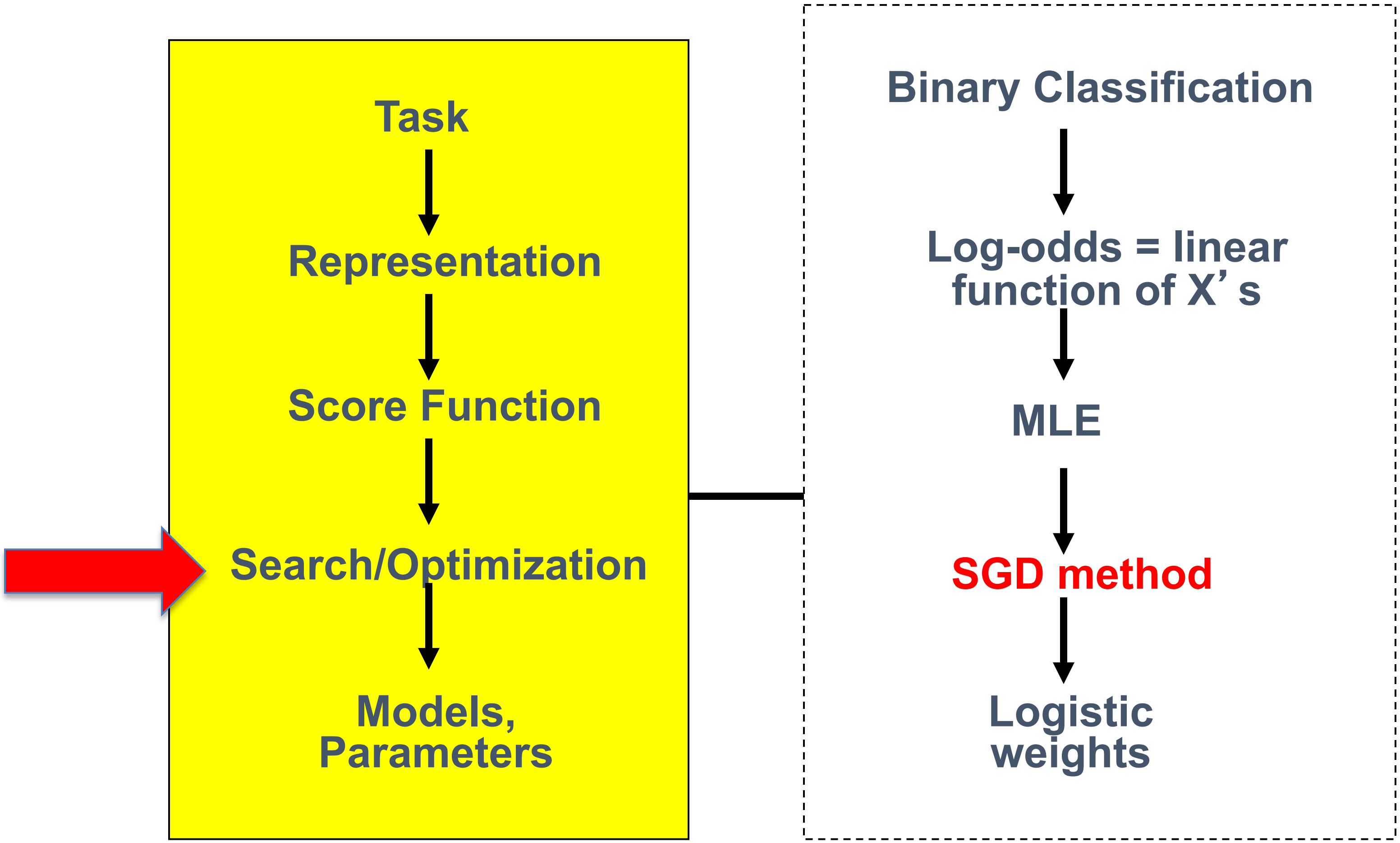
$$= \sum_{i=1}^{N} \{y_i \log(\Pr(Y=1 \mid X = x_i)) + (1 - y_i)\log(\Pr(Y=0 \mid X = x_i))\}$$

$$= \sum_{i=1}^{N} (y_i \log \frac{\exp(\beta^T x_i)}{1+\exp(\beta^T x_i)}) + (1-y_i)\log \frac{1}{1+\exp(\beta^T x_i)})$$

$$= \sum_{i=1}^{N} (y_i \beta^T x_i - \log(1+\exp(\beta^T x_i)))$$

See Extra Slides How to used Newton-Raphson optimization

Logistic Regression



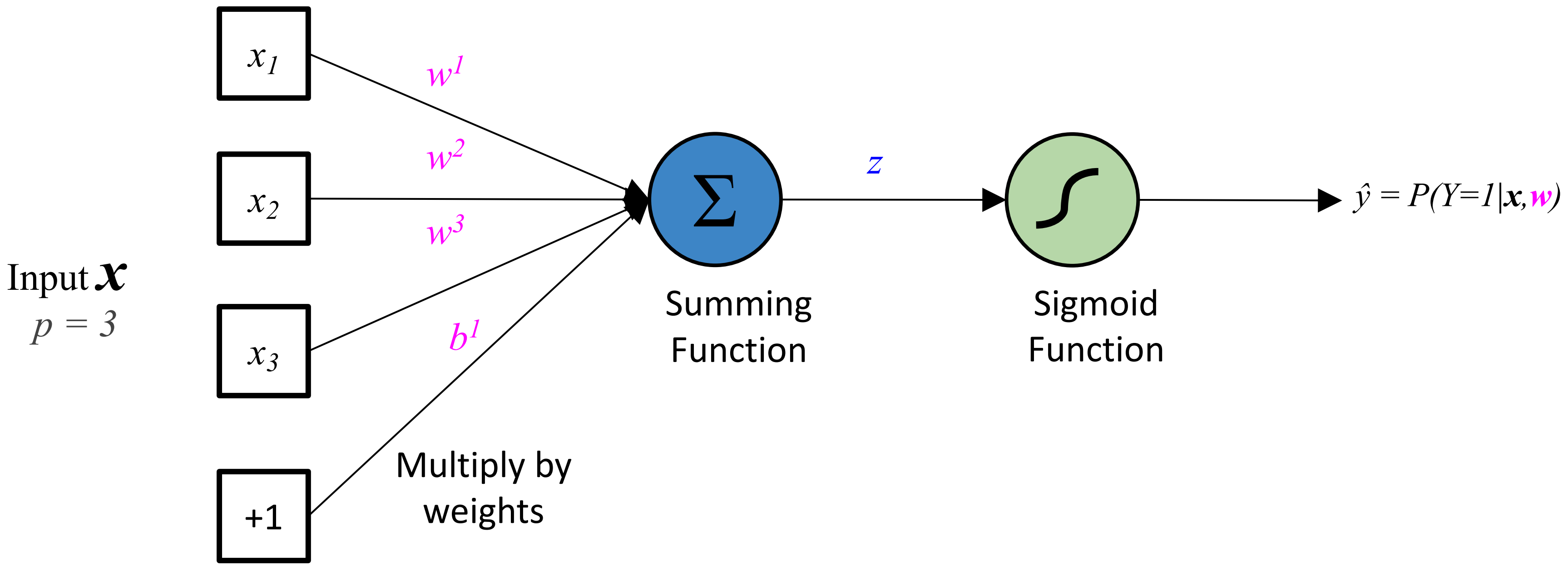
$$P(y=1|x) = \frac{e^{\beta_0+\beta^T x}}{1+e^{\beta_0+\beta^T x}}$$

ReWrite Logistic Regression as two stages:

First:
Summing $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$

Second:
Sigmoid
Squashing $\hat{y} = P(y=1|x) = \frac{e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p}} = \frac{e^z}{1 + e^z}$

One “Neuron”: Block View of Logistic Regression

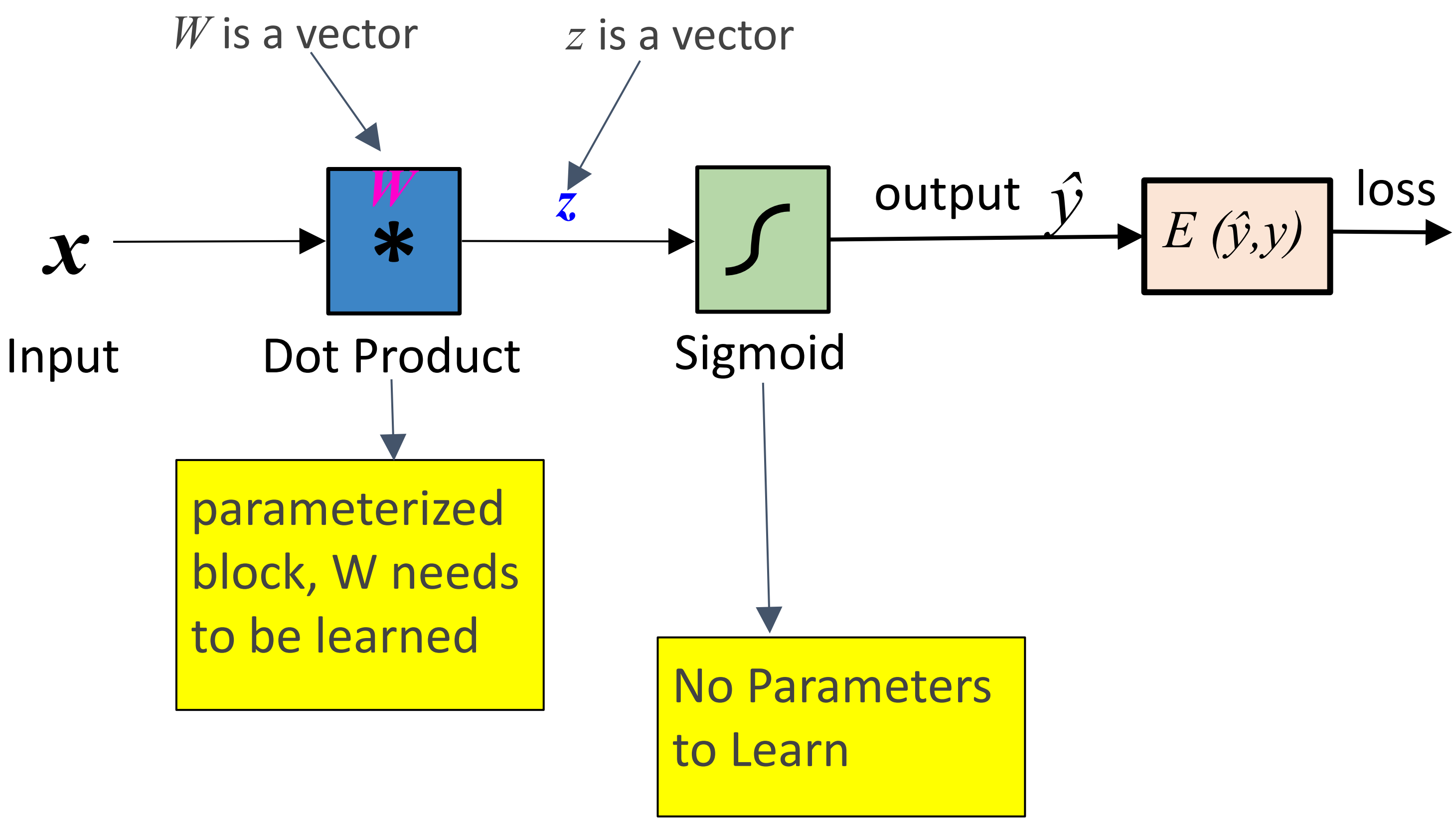


$$z = \boldsymbol{w}^T \cdot \boldsymbol{x} + \boldsymbol{b}$$

$$y = sigmoid(z)$$

$$= \frac{e^z}{1 + e^z}$$

e.g., "Block View" of Logistic Regression



Review: Stochastic GD ➔

- For LR: linear regression, We have the following gradient descent rule:

$$\theta_j^{t+1} = \theta_j^t - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \Big|_t$$

- ➔ For neural network, we have the delta rule

$$\Delta \mathrm{w} = -\eta \frac{\partial E}{\partial W^t}$$

$$W^{t+1} = W^t - \eta \frac{\partial E}{\partial W^t} = W^t + \Delta w$$

Logistic Regression

Task	Binary Classification
Representation	Log-odds = linear function of X's
Score Function	MLE
Search/Optimization	Iterative (Newton) method / SGD
Models, Parameters	Logistic weights

$$P(y=1|x) = \frac{e^{\beta_0+\beta^T x}}{1+e^{\beta_0+\beta^T x}}$$

Three major sections for classification

- We can divide the large variety of classification approaches into roughly three major types

1. Discriminative

    directly estimate a decision rule/boundary

    e.g~~., support vector machine~~, decision tree, ~~logistic regression,~~

    e.g. neural networks (NN), deep NN

2. Generative:

    build a generative statistical model

    e.g., Bayesian networks, Naïve Bayes classifier

3. Instance based classifiers

    - Use observation directly (no models)

    ~~- e.g. K nearest neighbors~~

References

- ❑ Prof. Tan, Steinbach, Kumar's "Introduction to Data Mining" slide
- ❑ Prof. Andrew Moore's slides
- ❑ Prof. Eric Xing's slides
- ❑ Prof. Ke Chen NB slides
- ❑ Hastie, Trevor, et al. *The elements of statistical learning*. Vol. 2. No. 1. New York: Springer, 2009.