Linear Classifiers

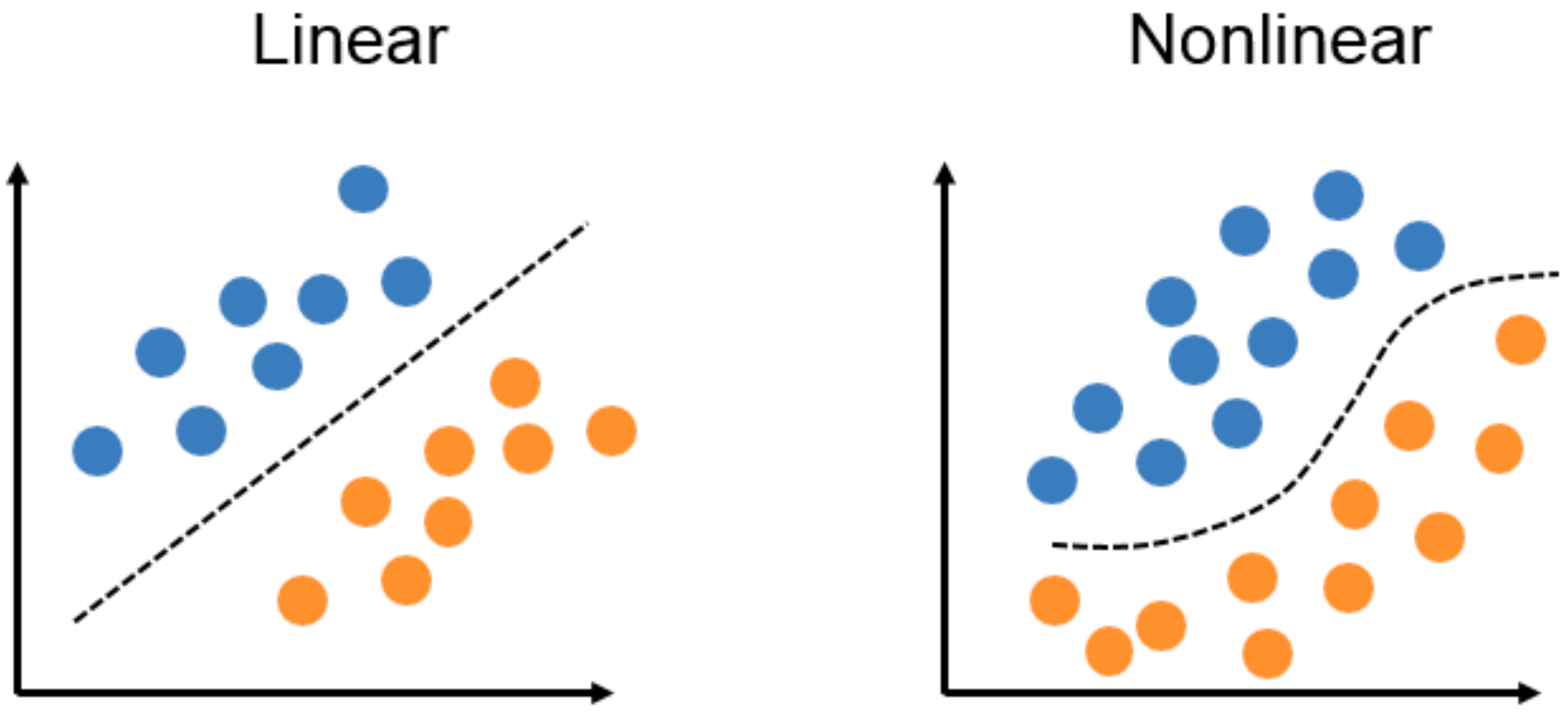
Machine Learning and Deep Learning
Lesson #5

AImage Lab
University of Modena and Reggio Emilia

Linear Classification

- What is meant by linear classification?

input)



Linear Classification…

- There is a discriminant function $\delta_k(x)$ for each class k

- Classification rule: $$R_k = \{x : k = \arg\max_j \delta_j(x)\}$$

- In higher dimensional space the decision boundaries are piecewise hyperplanar

- Remember that 0-1 loss function led to the classification rule:

$$R_k = \{x : k = \arg\max_j P(G = j \mid X = x)\}$$

- So, $P(G = k \mid X)$ can serve as $\delta_k(x)$

Linear Classification…

- All we require here is the class boundaries $\{x:\delta_k(x) = \delta_j(x)\}$ be linear for every (k, j) pair
- One can achieve this if $\delta_k(x)$ themselves are linear or any monotone transform of $\delta_k(x)$ is linear
  - An example:

$$P(G = 1 \mid X = x) = \frac{\exp(\beta_0 + \beta^T x)}{1 + \exp(\beta_0 + \beta^T x)}$$

$$P(G = 2 \mid X = x) = \frac{1}{1 + \exp(\beta_0 + \beta^T x)}$$

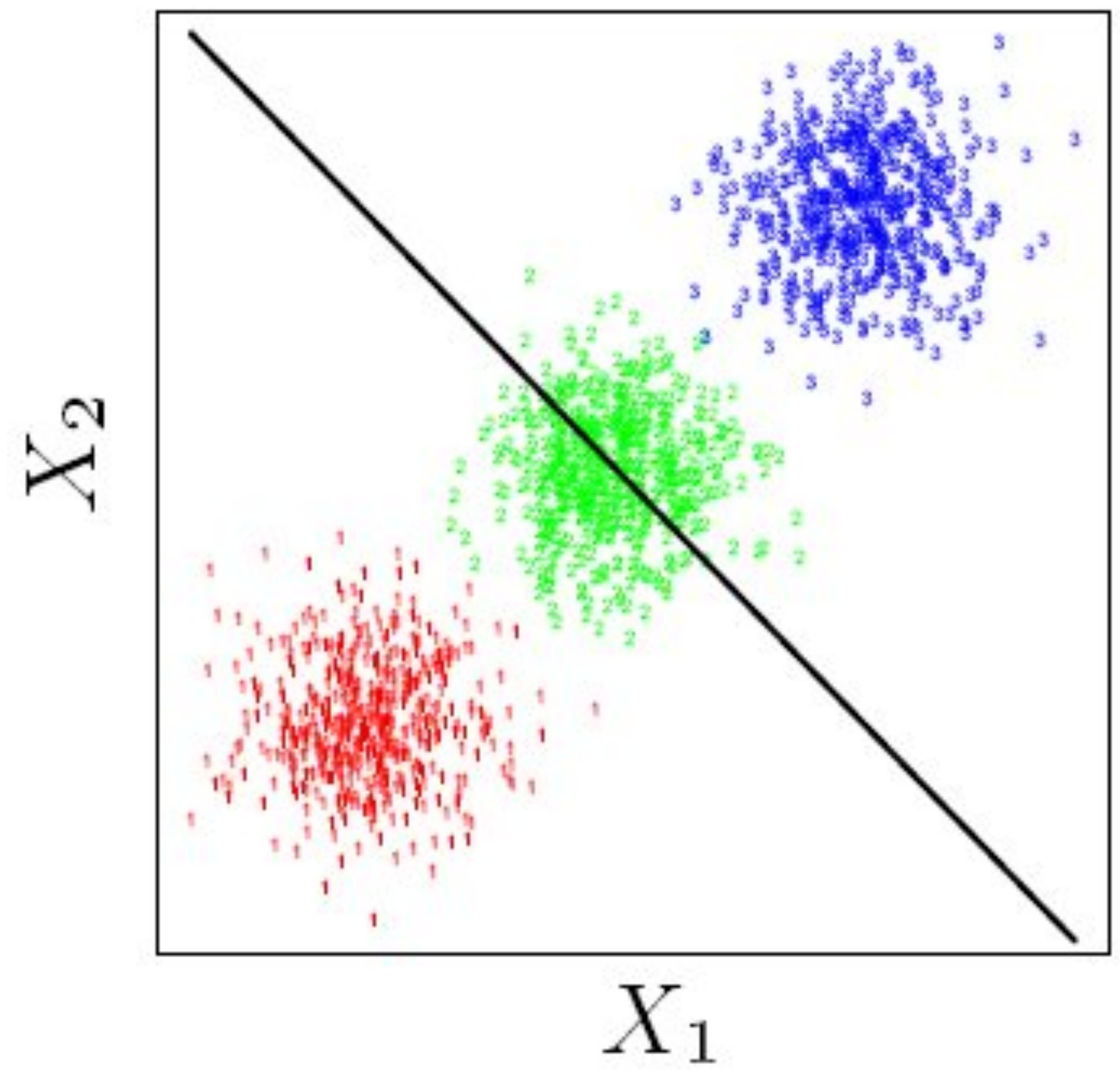
$$\log\left[\frac{P(G = 1 \mid X = x)}{P(G = 2 \mid X = x)}\right] = \beta_0 + \beta^T x$$

Linear

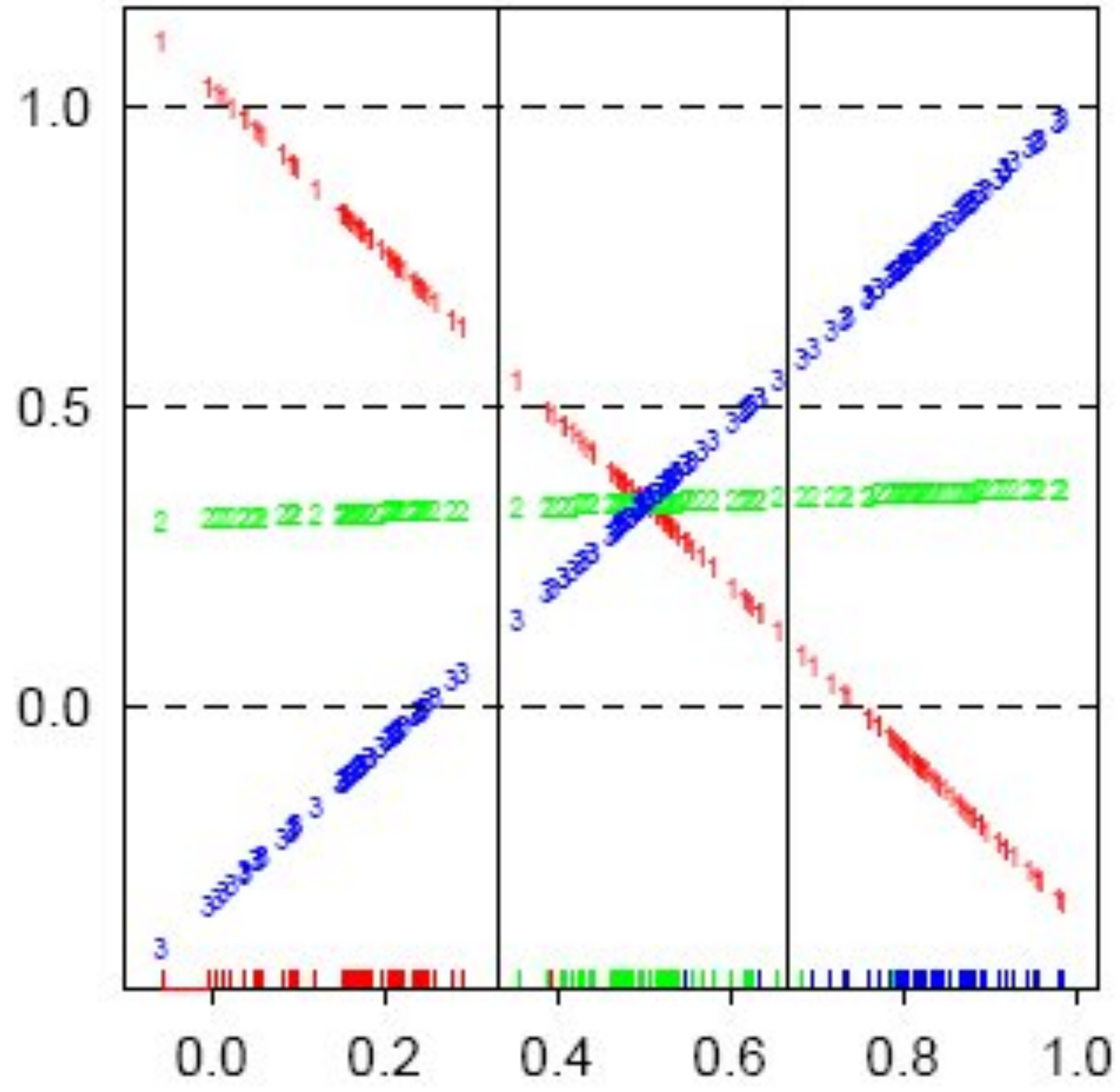
The Masking

Linear regression of the indicator matrix can lead to masking

2D input space and three classes



Masking



Viewing direction

LDA can avoid this masking

Linear Discriminant Analysis

Linear Discriminant Analysis

Essentially **minimum error Bayes' classifier**

- Assumes that the conditional class densities are (multivariate) Gaussian
- Assumes equal covariance matrix for every class

Posterior probability $$\Pr(G = k \mid X = x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^{K} f_l(x)\pi_l}$$

Application of Bayes rule

π_k is the prior probability for class k

$f_k(x)$ is class conditional density or likelihood density

$$f_k(x) = \frac{1}{(2\pi)^{p/2} |\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^T \boldsymbol{\Sigma}^{-1}(x-\mu_k)\right)$$

LDA…Continues

$$\log \frac{\Pr(G=k \mid X=x)}{\Pr(G=l \mid X=x)} = \log \frac{\pi_k}{\pi_l} + \log \frac{f_k}{f_l}$$

$$= \underbrace{(\log \pi_k + x^T \mathbf{\Sigma}^{-1} \mu_k - \frac{1}{2} \mu_k^T \mathbf{\Sigma}^{-1} \mu_k)}_{\delta_k(x)} - \underbrace{(\log \pi_l + x^T \mathbf{\Sigma}^{-1} \mu_l - \frac{1}{2} \mu_l^T \mathbf{\Sigma}^{-1} \mu_l)}_{\delta_l(x)}$$

Classification rule: $\hat{G}(x) = \arg\max_k \delta_k(x)$

is equivalent to: $\hat{G}(x) = \arg\max_k \Pr(G=k \mid X=x)$

The good old Bayes classifier!

LDA and training data

When are we going to use the training data?

DATASET:

- Total N input-output pairs
- N_k number of pairs in class k
- Total number of classes: K

$$(g_i, x_i),\ i = 1 : N$$

Training data used to estimate

1. Prior probabilities:

$$\hat{\pi}_k = N_k / N$$

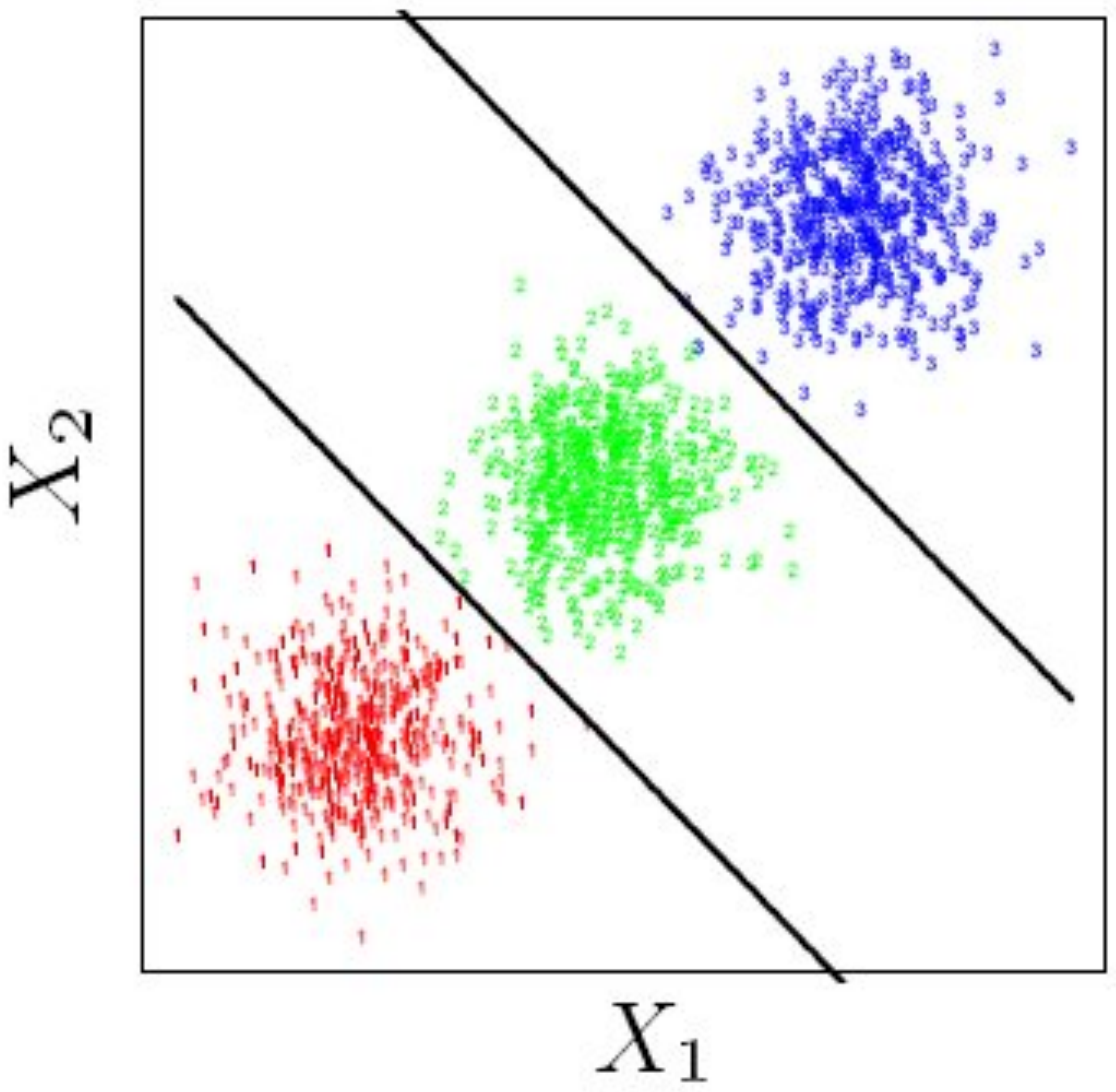
2. **Means**:

$$\hat{\mu}_k = \sum_{g_i = k} x_i / N_k$$

3. **Covariance matrix:**

$$\hat{\Sigma} = \sum_{k=1}^{K} \sum_{g_i} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T / (N - K)$$

LDA: Example



LDA was able to avoid masking here

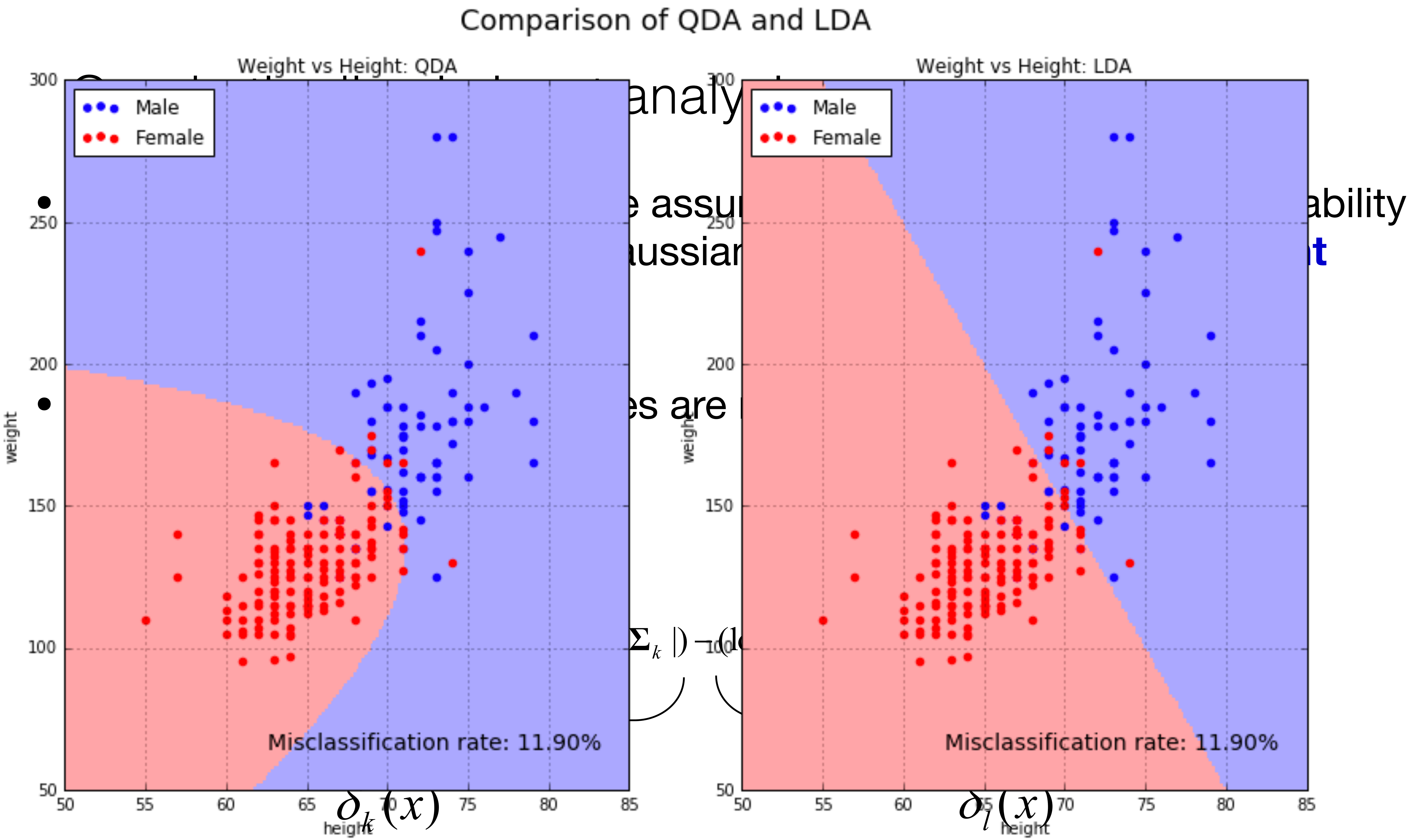
Quadratic discriminant analysis

- Relaxes the same covariance assumption– class conditional probability densities (still multivariate Gaussians) are allowed to have **different covariant matrices**

- The class decision boundaries are not linear rather **quadratic**

$$\log \frac{\Pr(G=k \mid X=x)}{\Pr(G=l \mid X=x)} = \log \frac{\pi_k}{\pi_l} + \log \frac{f_k}{f_l} =$$

$$\underbrace{(\log \pi_k - \frac{1}{2}(x-\mu_k)^T \mathbf{\Sigma}_k^{-1}(x-\mu_k) - \frac{1}{2}\log|\mathbf{\Sigma}_k|)}_{\delta_k(x)} - \underbrace{(\log \pi_l - \frac{1}{2}(x-\mu_l)^T \mathbf{\Sigma}_l^{-1}(x-\mu_l) - \frac{1}{2}\log|\mathbf{\Sigma}_l|)}_{\delta_l(x)}$$


Comparison of QDA and LDA
Weight vs Height: QDA
Male
Female
Misclassification rate: 11.90%
weight
height
300
250
200
150
100
50
50
55
60
65
70
75
80
85
Weight vs Height: LDA
Male
Female
Misclassification rate: 11.90%
weight
height
300
250
200
150
100
50
50
55
60
65
70
75
80
85

Logistic regression

Logistic Regression

- The output of regression is the posterior probability *i.e.*, Pr(output | input)
- Always ensures that the sum of output variables is 1 and each output is non-negative
- A linear classification method
- We need to know about two concepts to understand logistic regression
  - **Maximum likelihood estimation**
  - **Gradient Descent**

Logistic Regression Model

The method directly models the posterior probabilities as the output of regression

$$\Pr(G=k \mid X=x) = \frac{\exp(\beta_{k0}+\beta_k^T x)}{1+\sum_{l=1}^{K-1}\exp(\beta_{l0}+\beta_l^T x)},$$

$$\Pr(G=K \mid X=x) = \frac{1}{1+\sum_{l=1}^{K-1}\exp(\beta_{l0}+\beta_l^T x)}$$

Note that the class boundaries are linear

- x is p-dimensional input vector
- β_k is a p-dimensional vector for each k
- Total number of parameters is $(K-1)(p+1)$

Logistic Regression Computation

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

Training set: (x_i, g_i), i=1,…,N

Log-likelihood:

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

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

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

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

x_i are $(p+1)$-dimensional input vector with leading entry 1
β is a $(p+1)$-dimensional vector
y_i = 1 if g_i =1; y_i = 0 if g_i =2

We want to maximize the log-likelihood in order to estimate β

Logistic Regression Computation

Log-likelihood:

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

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

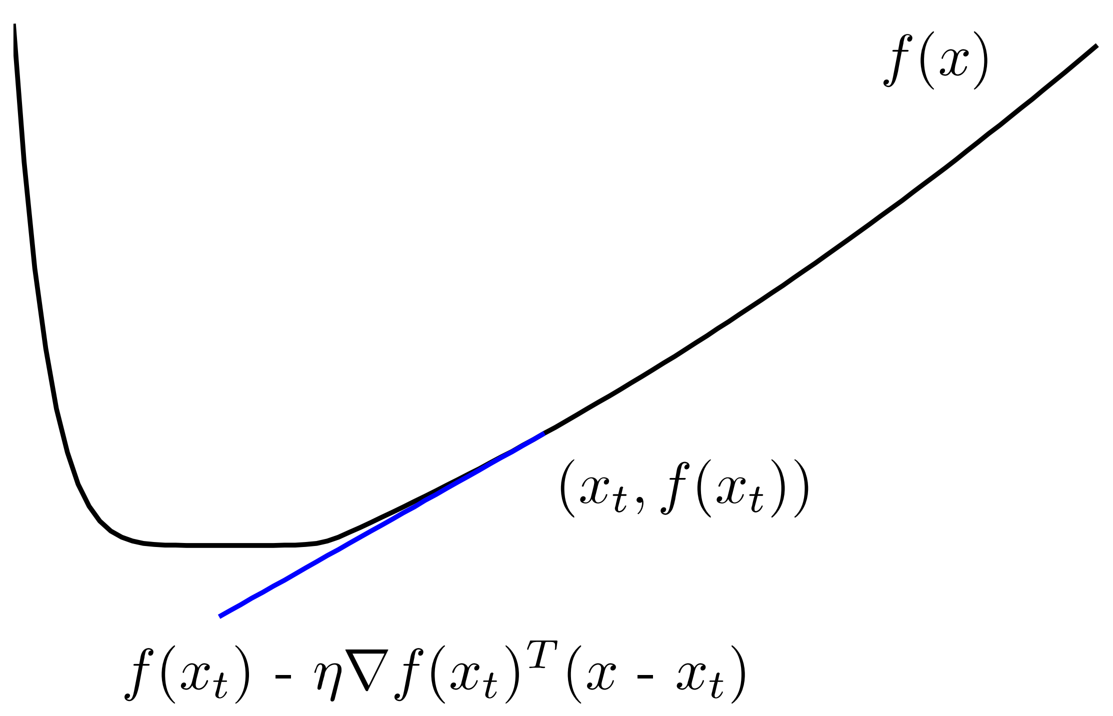
Log Likelyhood assume the form of the **negative Binary cross Entropy** on the Dataset

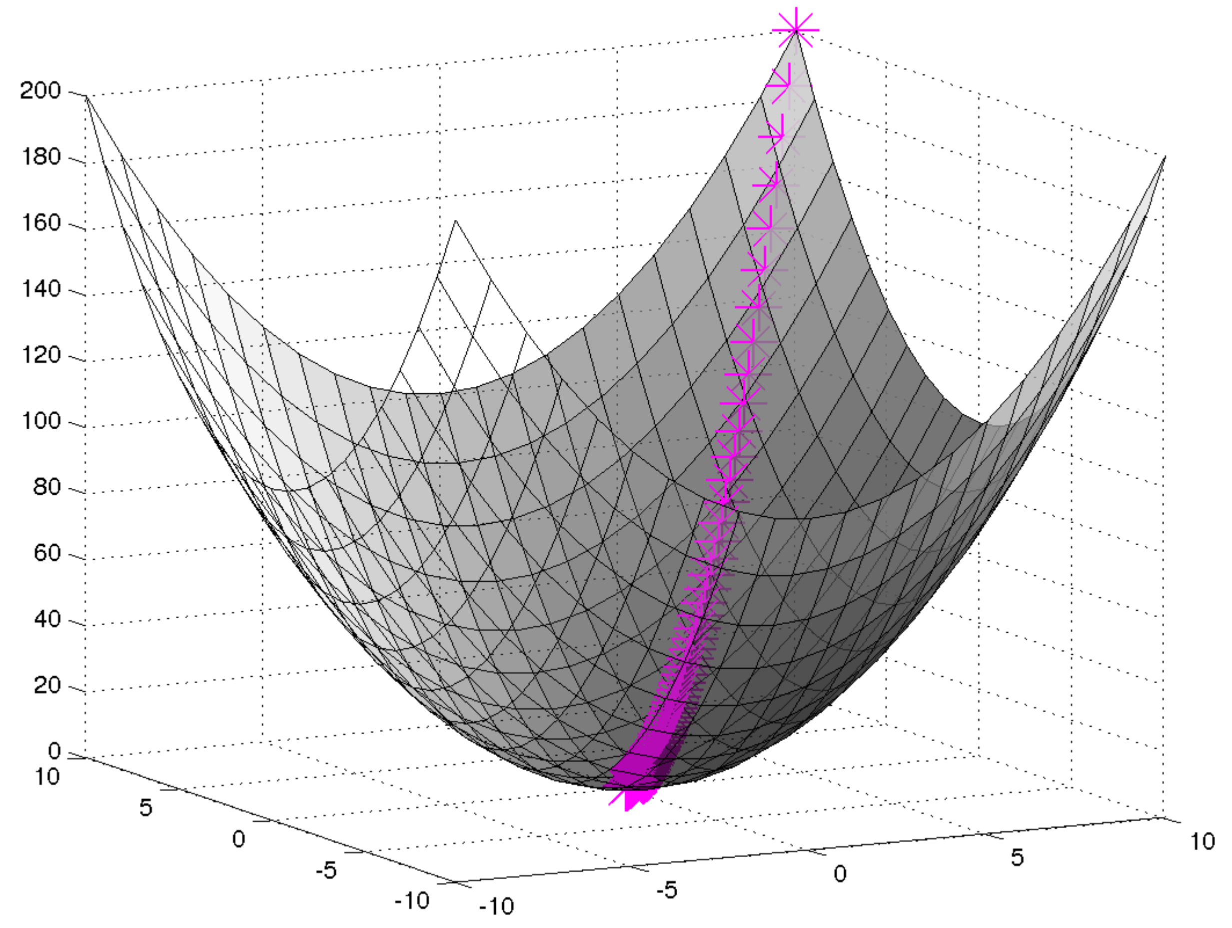
Maximize the likelihood is then minimize the cross entropy

We want to maximize the log-likelihood in order to estimate β

Gradient Descent

- **Iterative Minimization** technique based on local derivative.
- Define objectives as a **Loss Function** and Minimize it
- Lead to a **global minimum** iff the function to optimize is convex
- Use single steps of the form $x_{t+1} = x_t - \eta_t \nabla f(x_t)$
- To optimize





Gradient Descent Algorithm

Minimize f(x) w.r.t. x with f(x) convex

1. **Initial** condition : Pick x_0 at random

2. **Iterate** (while t in 0…..untill convergence):

    1. Compute the Gradient of f(x) at x_t

    2. Compute $x_{t+1} = x_t - \eta_t \nabla f(x_t)$

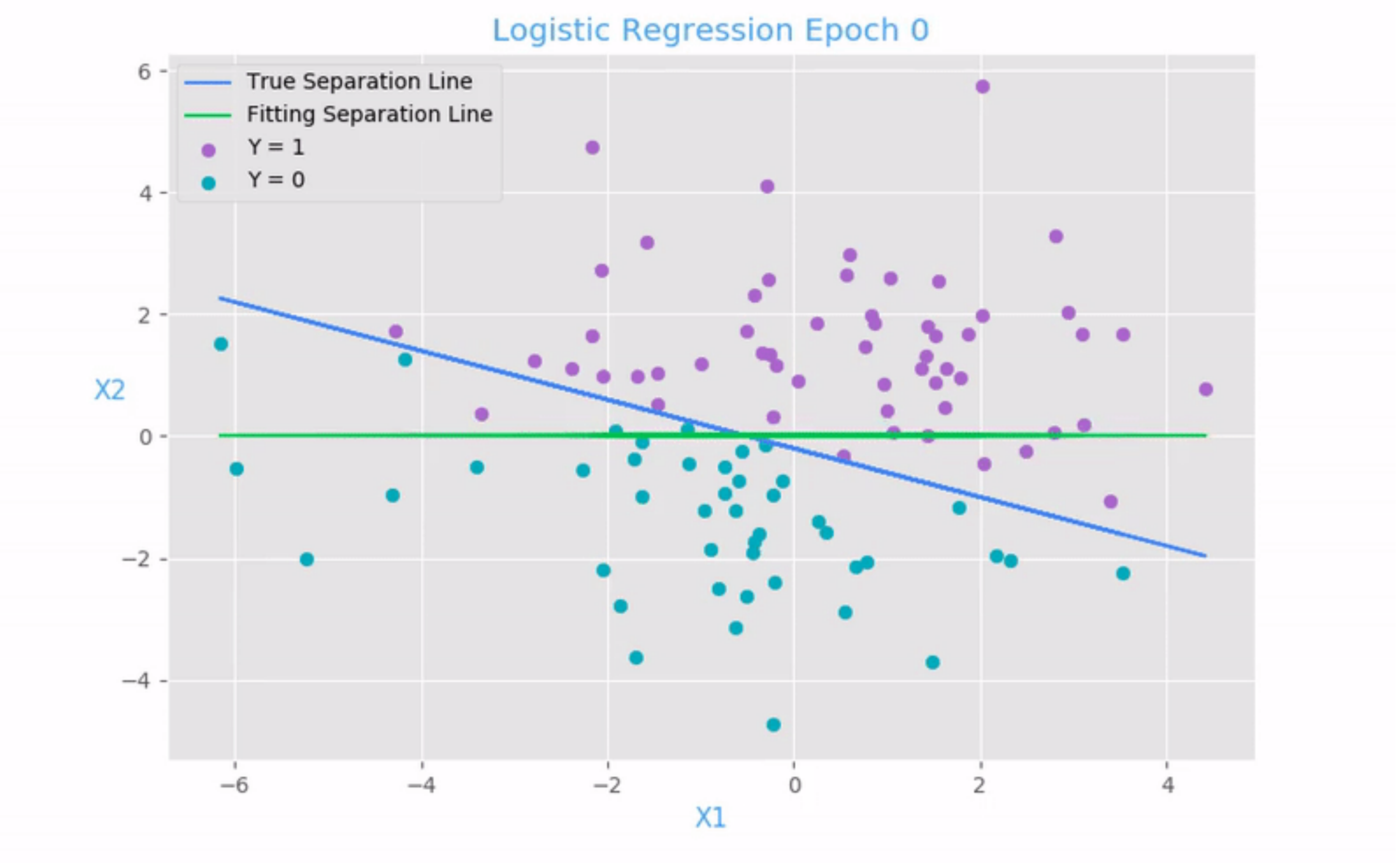
    3. Check if then STOP else STOP if MAXIMUM iterations reached

3. **Output** $x_{convergence}$

Gradient descent for logistic regression

Objective:

- Minimize the binary cross entropy on the Training set (x_i, y_i) with i=1…N



$$min_\beta L = \sum_{i=1}^{N} y_i log(P(G=1|x_i) + (1-y_i)P(G=0|x_i)$$

$$with\ P(G=1|x_i) = \frac{\exp^{\beta^T x_i}}{1+exp\beta^T x_i}\ and$$

$$P(G=0|x_i) = \frac{1}{1+exp^{\beta^T x_i}}$$

- Taking derivative w.r.t. parameters

$$\frac{\partial L}{\partial \beta_j} = (y_i - \frac{\exp^{\beta^T x_i}}{1+exp^{\beta^T x_i}})x_j$$

- and the update rule follows

$$\beta_j^{t+1} = \beta_j^t - \eta \frac{\partial L}{\partial \beta_j}$$

Gradient descent for logistic regression

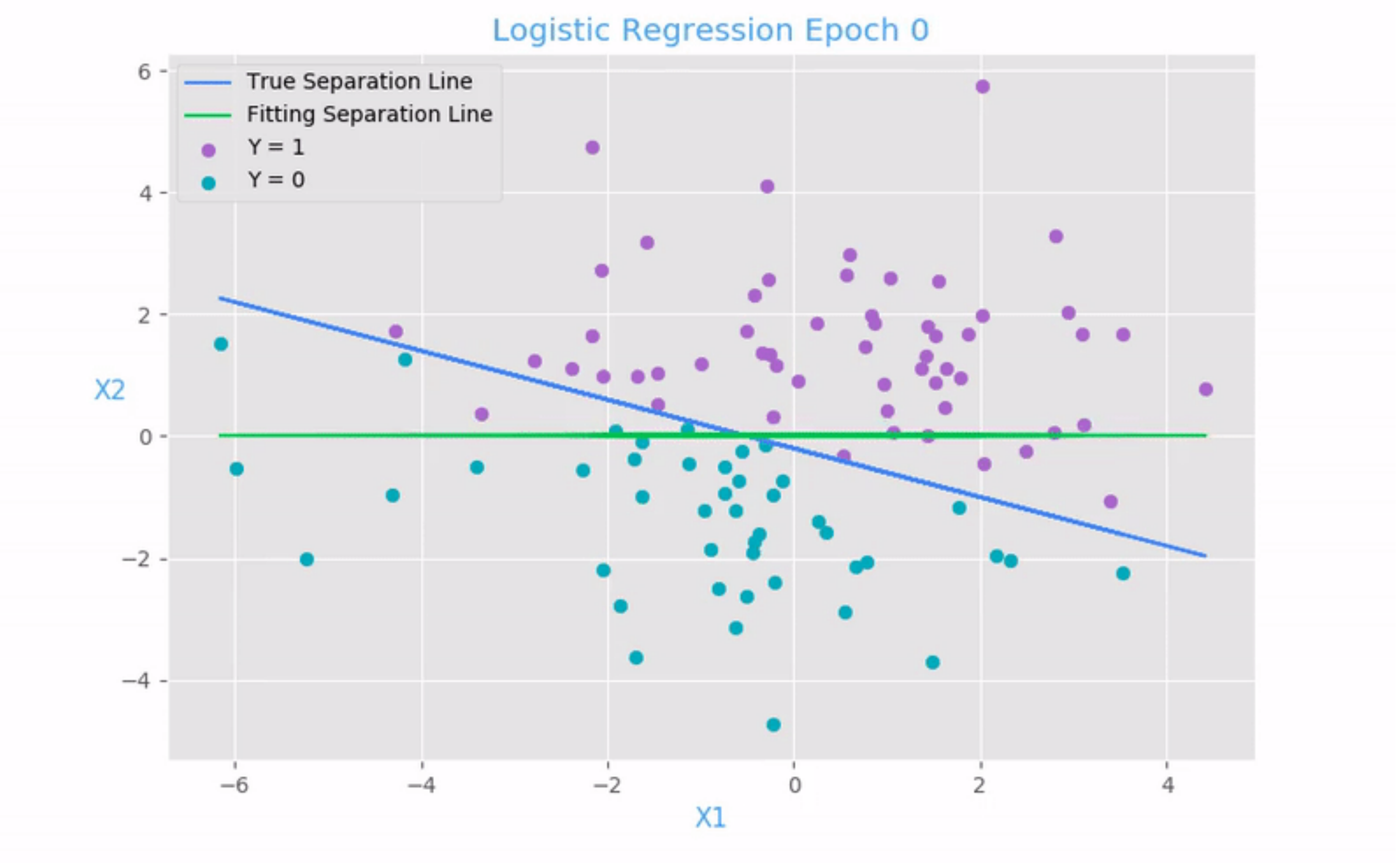
Objective:

- Minimize the binary cross entropy on the Training set (x_i, y_i) with i=1…N

$$min_\beta L = \sum_{i=1}^{N} y_i log(P(G=1|x_i) + (1-y_i)P(G=0|x_i)$$

$$with\ P(G=1|x_i) = \frac{\exp^{\beta^T x_i}}{1 + exp\beta^T x_i}\ and$$

$$P(G=0|x_i) = \frac{1}{1 + exp^{\beta^T x_i}}$$



- Taking derivative w.r.t. parameters

$$\frac{\partial L}{\partial \beta_j} = (y_i - \frac{\exp^{\beta^T x_i}}{1 + exp^{\beta^T x_i}})x_j$$

- and the update rule follows

$$\beta_j^{t+1} = \beta_j^t - \eta \frac{\partial L}{\partial \beta_j}$$

Example: South African Heart Disease

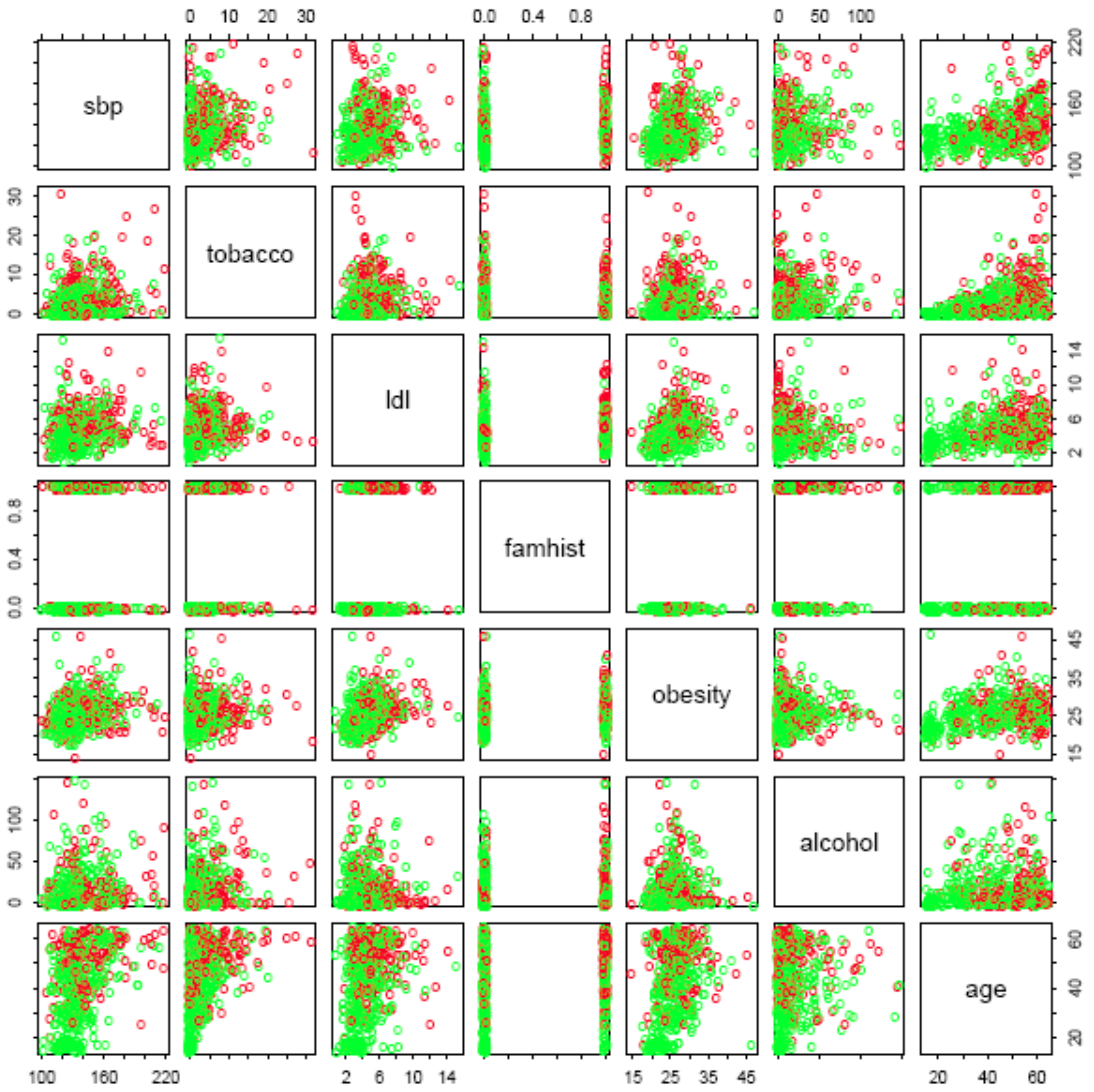


Figure 4.12: *A scatterplot matrix of the South African heart disease data. Each plot shows a pair of risk factors, and the cases and controls are color coded (red is a case). The variable* family history of heart disease (**famhist**) *is binary (yes or no).*

Example: South African Heart Disease…

After data fitting in the logistic regression model:

$$\Pr(MI = yes \mid x) = \frac{\exp(-4.130 + 0.006x_{\text{sbp}} + 0.08x_{\text{tobaco}} + 0.185x_{\text{ldl}} + 0.939x_{\text{famhist}} - 0.035x_{\text{obesity}} + 0.001x_{\text{alcohol}} + 0.043x_{\text{age}})}{1 + \exp(-4.130 + 0.006x_{\text{sbp}} + 0.08x_{\text{tobaco}} + 0.185x_{\text{ldl}} + 0.939x_{\text{famhist}} - 0.035x_{\text{obesity}} + 0.001x_{\text{alcohol}} + 0.043x_{\text{age}})}$$

	Coefficient	Std. Error	Z Score
(Intercept)	-4.130	0.964	-4.285
sbp	0.006	0.006	1.023
tobacco	0.080	0.026	3.034
ldl	0.185	0.057	3.219
famhist	0.939	0.225	4.178
obesity	-0.035	0.029	-1.187
alcohol	0.001	0.004	0.136
age	0.043	0.010	4.184

Example: South African Heart Disease…

After ignoring negligible coefficients:

$$\Pr(MI = yes \mid x) = \frac{\exp(-4.204 + 0.081x_{\text{tobaco}} + 0.168x_{\text{ldl}} + 0.924x_{\text{famhist}} + 0.044x_{\text{age}})}{1 + \exp(-4.204 + 0.081x_{\text{tobaco}} + 0.168x_{\text{ldl}} + 0.924x_{\text{famhist}} + 0.044x_{\text{age}})}$$

What happened to systolic blood pressure? Obesity?

LDA vs. Logistic Regression

- LDA (Generative model)
  - Assumes Gaussian class-conditional densities and a common covariance
  - Model parameters are estimated by maximizing the full log likelihood, parameters for each class are estimated independently of other classes, $Kp+p(p+1)/2+(K-1)$ parameters
  - Makes use of marginal density information $\Pr(X)$
  - Easier to train, low variance, more efficient if model is correct
  - Higher asymptotic error, but converges faster

- Logistic Regression (**Discriminative model**)
  - Assumes class-conditional densities are members of the (same) exponential family distribution
  - Model parameters are estimated by maximizing the conditional log likelihood, simultaneous consideration of all other classes, $(K-1)(p+1)$ parameters
  - Ignores marginal density information $\Pr(X)$
  - Harder to train, robust to uncertainty about the data generation process
  - Lower asymptotic error, but converges more slowly

Generative vs. Discriminative Learning

	Generative	**Discriminative**
Example	Linear Discriminant Analysis	Logistic Regression
Objective Functions	Full log likelihood: $\sum_i \log p_\theta(x_i, y_i)$	Conditional log likelihood $\sum_i \log p_\theta(y_i \mid x_i)$
Model Assumptions	Class densities: $p(x \mid y = k)$ e.g. Gaussian in LDA	Discriminant functions $\lambda_k(x)$
Parameter Estimation	"Easy" – One single sweep	"Hard" – iterative optimization
Advantages	More efficient if model correct, borrows strength from $p(x)$	More flexible, robust because fewer assumptions
Disadvantages	Bias if model is incorrect	May also be biased. Ignores information in $p(x)$