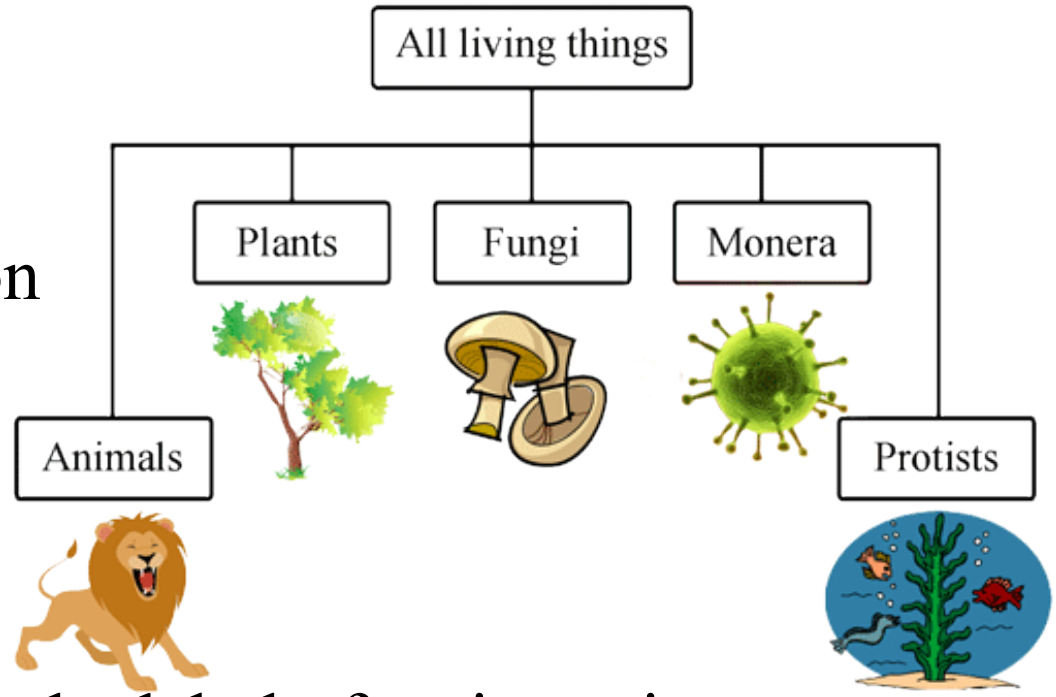
CSI 436/536
Introduction to Machine Learning

Classification and discriminative LLSE

Professor Siwei Lyu
Computer Science
University at Albany, State University of New York

Classification problem

- In many real life problems, we need to map input data to different categories, and this is known as classification
- Classification is one basic machine learning problem
  - Medical diagnosis
  - Spam filtering
  - Face detection/recognition
  - Object recognition
  - Prompt word recognition "Hey Siri", "Alexa"



- The ML model that predicts the label of an input is known as the **classifier**

General classification problem

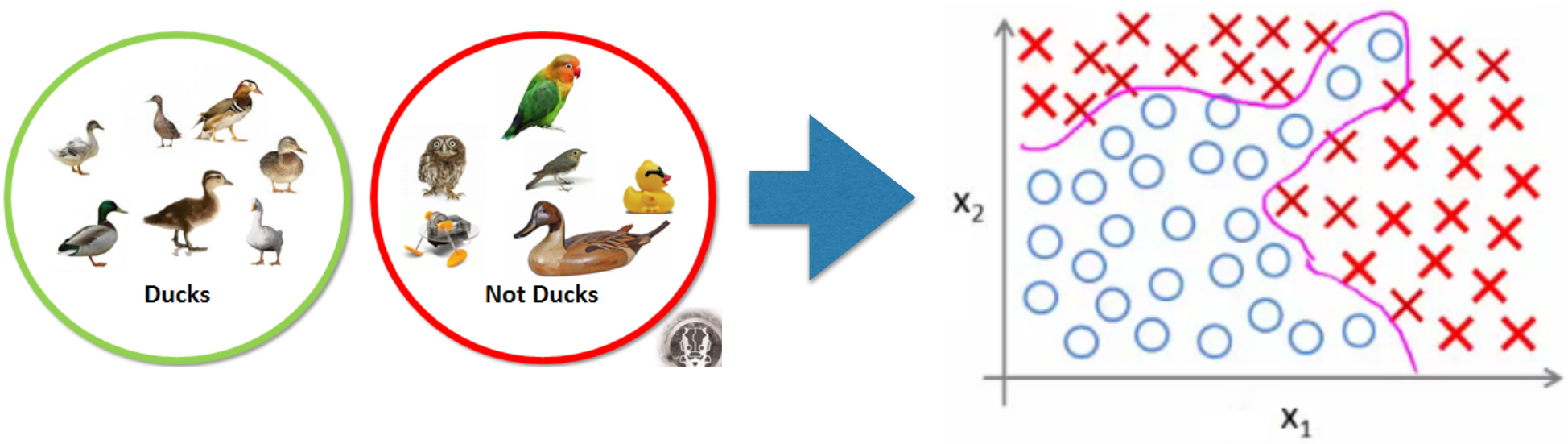
- Training
  - Training data matrix
    data points are column vectors

$$X = \begin{pmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_N \\ | & | & & | \end{pmatrix}$$

  - Training targets $y = (y_1, y_2, \cdots, y_N)^T \in \{b_1, b_2, \cdots, b_m\}^N$
    - Multi-label learning: an input can have many labels
  - parametric classifier $c_w(\cdot) : R^d \mapsto \{b_1, b_2, \cdots, b_m\}$
  - **loss function** $L(y - c_w(x)) \geq 0$
  - Numerical procedure to find optimal w to minimize the learning objective $\sum_{i=1}^{n} L(y_i - c_w(x_i))$
- In testing, for input x and generate prediction $c_w(x)$
  - **metric function** $m(y - c_w(x)) \geq 0$

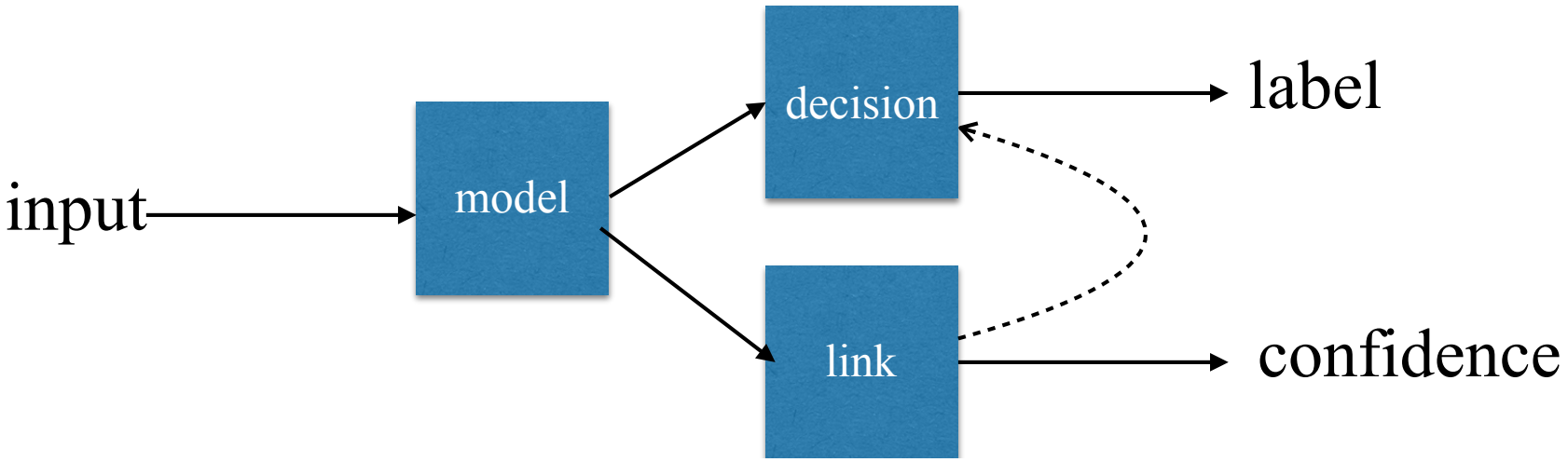
Binary classification

- When the class label only takes two values
  - Usually assume the class labels to take values (0,1) or (-1, +1) for convenience
  - The most widely studied problem in machine learning
  - Multi-class classification problem can be solved as sequences of binary classification problems



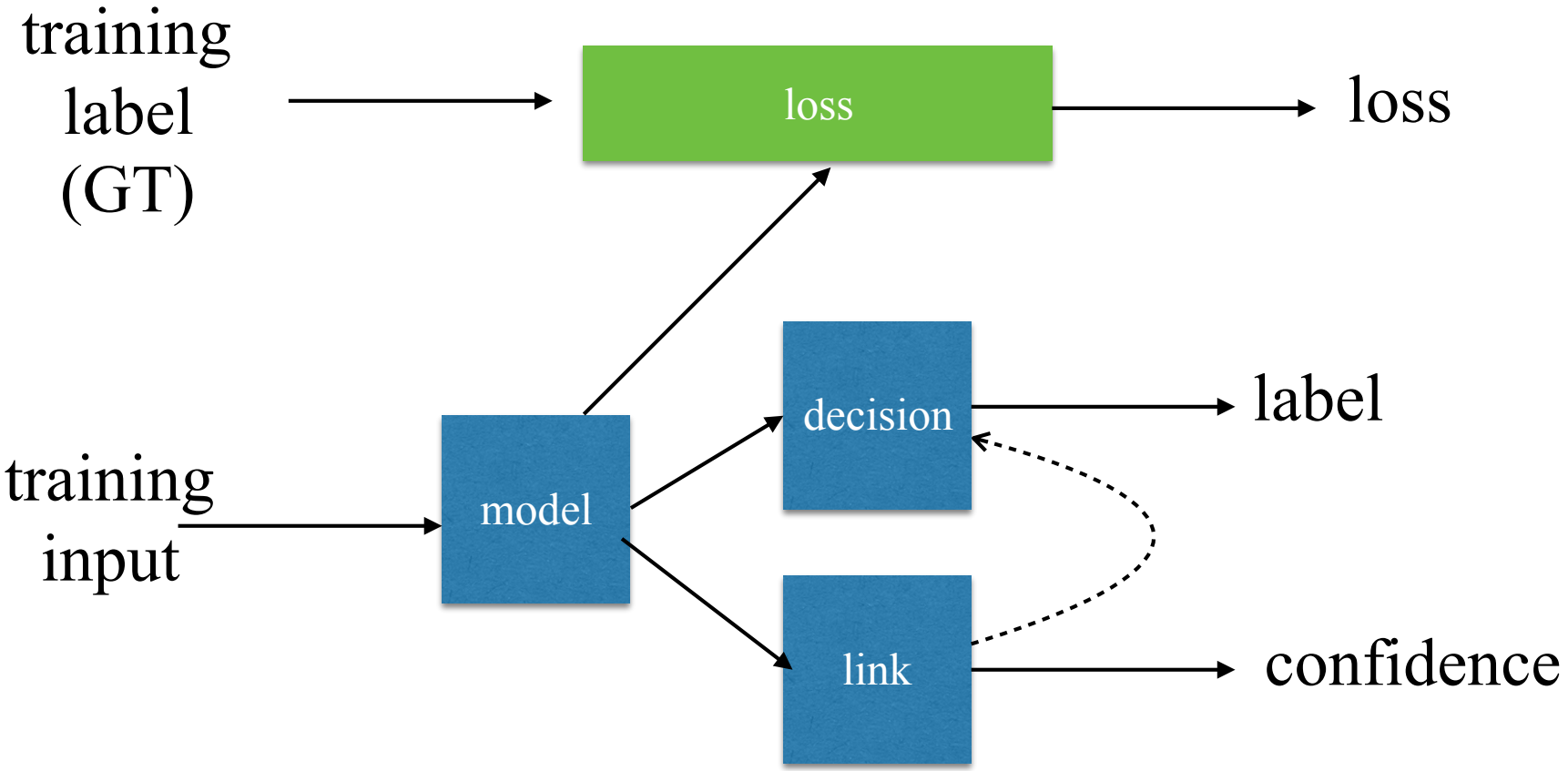
terminology

- prediction function (model): f
  - maps input to R, many choices
- link function (confidence): p = l(f)
  - maps R to [0,1], usually is the soft-max function
- decision function (classifier): c(f) or c(l(f))
  - maps R or [0,1] to {-1,+1}, usually is the sign function



training a binary classifier

- form a loss function $\ell(y, c) : \{-1, +1\} \times R \mapsto R_+$



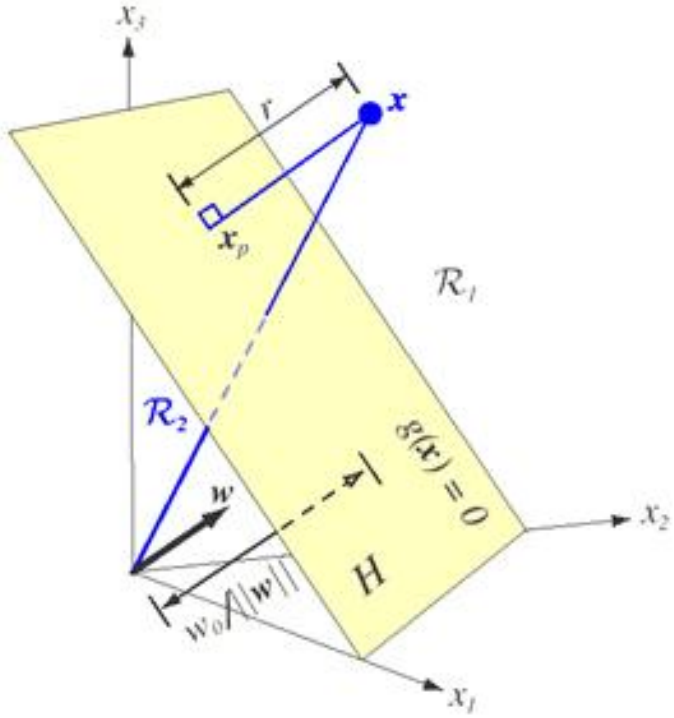
- minimize it over the training dataset

Prediction function

- Linear prediction function

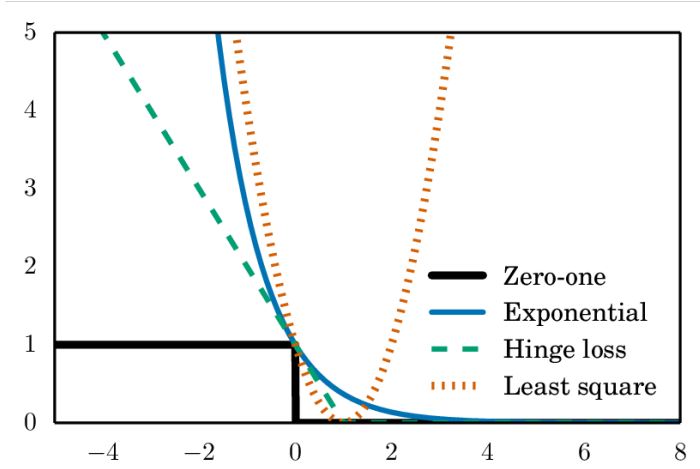
$$f_w(x) = w^\top x + c$$

  - Parameter w: surface normal
  - Parameter b: threshold
  - Decision surface f(x) = 0
- Nonlinear function
  - Radial basis functions
  - Kernel support vector machines
  - Decision trees
  - Boosted classifiers (piece-wise linear)
  - Neural networks



common individual loss functions

- 0-1 loss (metric) $1(yf(x) < 0)$ ideal case
- Use surrogate loss
  - margin loss, depending on $yf(x)$
  - **square loss** $(1- yf(x))^2$
  - **hinge loss** $\max(0,1-yf(x))$
  - **exponential loss** $\exp(-yf(x))$
  - logistic loss $\frac{1}{\log 2}\log(1+\exp(-yf(x)))$
  - squared hinge loss $\max(0,1-yf(x))^2$
- cross-entropy loss $-y \log p(x) - (1-y)\log(1-p(x))$
- Area under ROC curve (AUC) metric



LLSE classification problem

- Training
  - Training data matrix
    data points are column vectors
    $$X = \begin{pmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_N \\ | & | & & | \end{pmatrix}$$
  - Binary targets $y = (y_1, y_2, \cdots, y_N)^T \in \{-1, +1\}^N$
  - Linear classification function $c_w(x) = \text{sign}(w^T x + c)$
    - Absorb b by introducing a homogeneous transform
      $\tilde{x} = \begin{pmatrix} x \\ 1 \end{pmatrix}$ and $\tilde{w} = \begin{pmatrix} w \\ c \end{pmatrix}$
  - **Least squares loss function** $(y - w^T x)^2 = (1 - yw^T x)^2$
  - Numerical procedure to find optimal w to minimize the learning objective $\sum_{i=1}^{n} (y_i - w^T x_i)^2$
- In testing, for input x and generate prediction $f_w(x)$
  - 0-1 **metric function** $1_{yf_w(x)}$, note it is different from the loss function

Use LLSE to solve binary classification

- Basic idea: restrict target (y) to two choices {-b, a}, and find a linear decision function $f(x) = w^T x + c$, such that if x is in class 0, f(x) is closer to -b, and if x is in class 1, f(x) is closer to a
- We introduce data matrices

$$X_+ = \begin{pmatrix} | & | & & | \\ x_1^+ & x_2^+ & \cdots & x_{N_+}^+ \\ | & | & & | \end{pmatrix} \quad X_- = \begin{pmatrix} | & | & & | \\ x_1^- & x_2^- & \cdots & x_{N_-}^- \\ | & | & & | \end{pmatrix}$$

  $X = (X_+, X_-)$, and $N = N_+ + N_-$

- We also introduce all one vectors $1_{N_+}, 1_{N_-}$, and 1_N

Problem formulation

- We aim to minimize the LLSE objective function

$$\min_{w,c} \frac{1}{N}\|X_+^T w + c1_{N_+} - a1_{N_+}\|^2 + \frac{1}{N}\|X_-^T w + c1_{N_-} + b1_{N_-}\|^2$$

- We first minimize c

$$\frac{\partial}{\partial c}\|X_+^T w + c1_{N_+} - a1_{N_+}\|^2 = 2cN_+ + 21_{N_+}^T X_+^T w - 2aN_+$$

$$\frac{\partial}{\partial c}\|X_-^T w + c1_{N_-} + b1_{N_-}\|^2 = 2cN_- + 21_{N_-}^T X_-^T w + 2bN_-$$

summing terms and ignoring factor 2,

$$cN + 1_N^T X^T w - aN_+ + bN_- = 0$$

so $c = -m^T w + a\frac{N_+}{N} - b\frac{N_-}{N}$ where $m = \frac{1}{N}X1_N$ is the mean of all data

Intermediate result

- We can use optimal c value to simplify

$$X_+^T w + c1_{N_+} - a1_{N_+} = X_+^T w - m^T w 1_{N_+} - \frac{N_-(a+b)}{N} 1_{N_+}$$

$$X_-^T w + c1_{N_-} + b1_{N_-} = X_-^T w - m^T w 1_{N_-} + \frac{N_+(a+b)}{N} 1_{N_+}$$

- next, note that we have $m^T w 1_{N_+} = (1_{N_+} m^T) w$ and $m^T w 1_{N_-} = (1_{N_-} m^T) w$, we further have

$$X_+^T w + c1_{N_+} - a1_{N_+} = (X_+^T - 1_{N_+} m^T) w - \frac{N_-(a+b)}{N} 1_{N_+}$$

$$X_-^T w + c1_{N_-} + b1_{N_-} = (X_-^T - 1_{N_-} m^T) w + \frac{N_+(a+b)}{N} 1_{N_-}$$

Solving for w

- Now we minimize w and dropping factor 2, we get

$$\frac{\partial}{\partial w}\|X_+^T w + \cdots\|^2 = (X_+ - (1_{N_+} m^T)^T)(X_+^T - 1_{N_+} m^T)w - N_- 1_{N_+} X_+^T \frac{a+b}{N} - \frac{N_- N_+ (a+b)}{N} m$$

$$\frac{\partial}{\partial w}\|X_-^T w + \cdots\|^2 = (X_- - (1_{N_-} m^T)^T)(X_-^T - 1_{N_-} m^T)w + N_+ 1_{N_-} X_-^T \frac{a+b}{N} + \frac{N_- N_+ (a+b)}{N} m$$

- Introducing $m_+ = \frac{1}{N_+} X_+ 1_{N_+}$ and $m_- = \frac{1}{N_-} X_- 1_{N_-}$
- Recall covariance matrices

$$S = \frac{1}{N}(X - (1_N m^T)^T)(X^T - 1_N m^T)$$

- the total gradient is

$$Sw - \frac{N_+ N_- (a+b)}{N^2}(m_+ - m_-) = 0, \text{ or } \quad w \propto S^{-1}(m_+ - m_-)$$

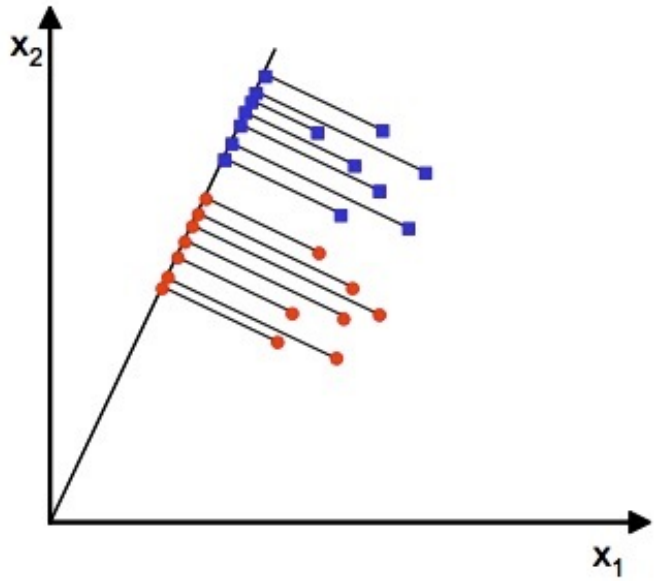
Discriminative LLSE

- LLSE solution to the binary classification problem gives the following linear decision function

$$f(x) = w^T x + c = \frac{N^2}{N_+ N_-(a+b)} S^{-1}(m_+ - m_-)^T(x - m) + a\frac{N_+}{N} - b\frac{N_-}{N}$$

  - Choosing a = N/N_+ and b= N/N_- leads to simpler decision function as

$$f(x) = w^T x + c = S^{-1}(m_+ - m_-)^T(x - m)$$



Summary

- The direction given by the discriminative LLSE will be exactly the same as the one given by linear discriminant analysis we will see later in the class, albeit the two are derived from different principles
- The parameters w and c are representations of the training data in X_+ and X_-, but they have much lower dimensionality (m+1), comparing the raw data (mN), they are usually called the ***features***
- We will cover more topics about binary classification in later part of the course
  - LDA, logistic regression, SVM, neural networks