CSCI 5521: Machine Learning Fundamentals

Linear Discrimination

Rui Kuang

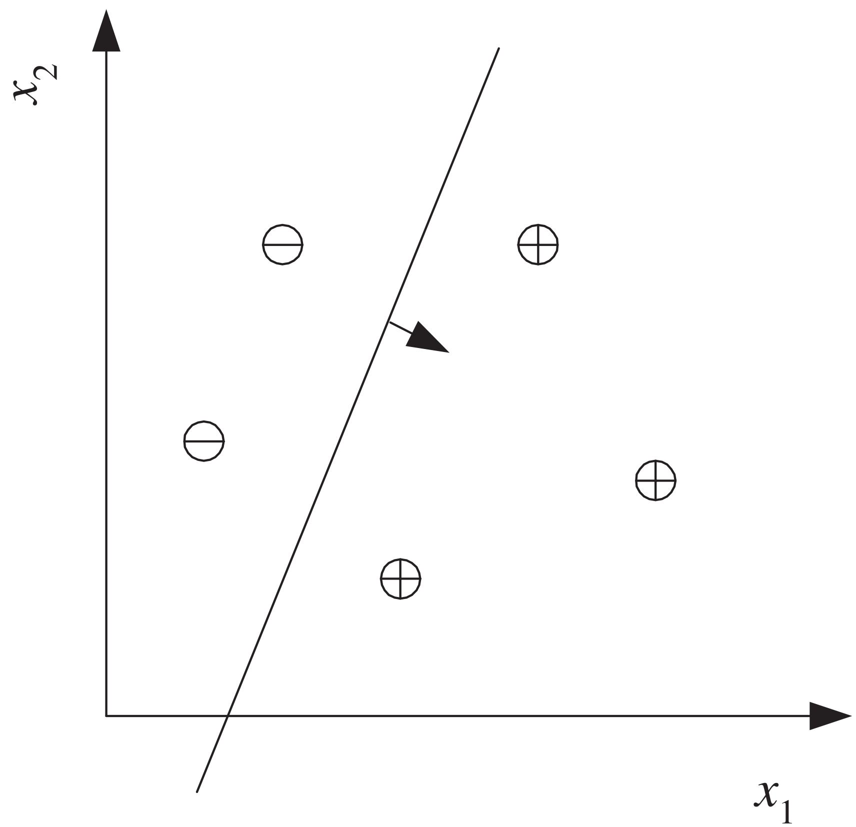
Department of Computer Science and Engineering

University of Minnesota

UNIVERSITY OF MINNESOTA

Twin Cities • Duluth • Morris • Crookston • Rochester • Other Locations

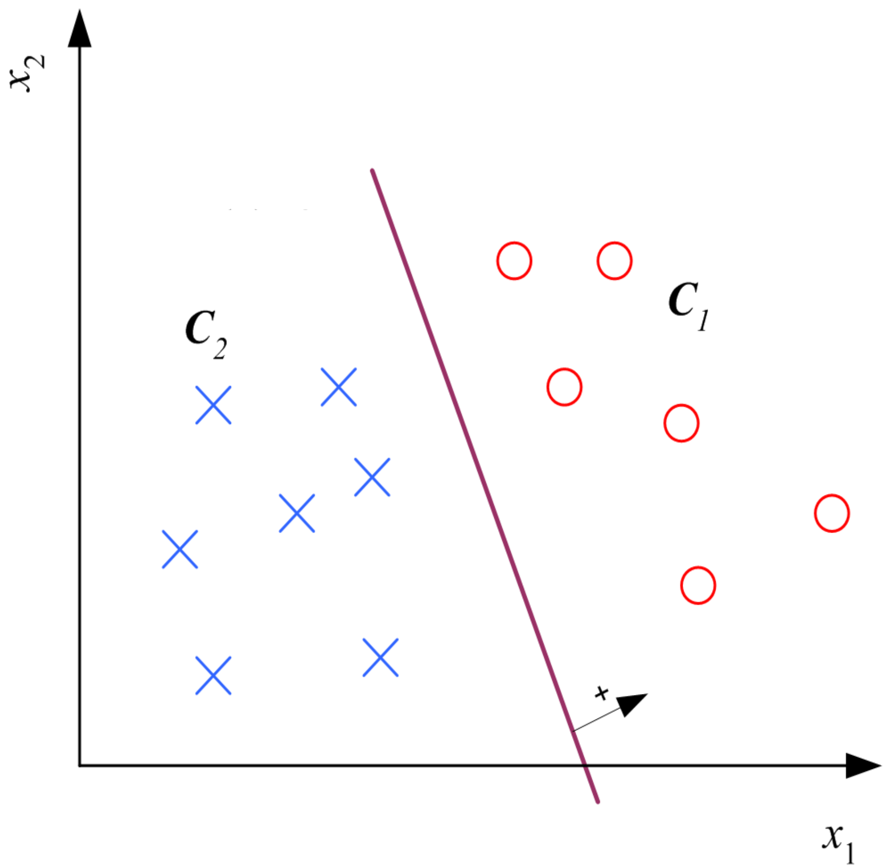
Linear Classifier



$h(x)=<w,x>+b$ is a linear classifier

$h(x)>0$ positive
$h(x)<0$ negative

Likelihood vs Discrimination



Likelihood/Posterior : $g_i(x) = p(x \mid C_i)P(C_i)$

Rely on estimation of the probability densities.

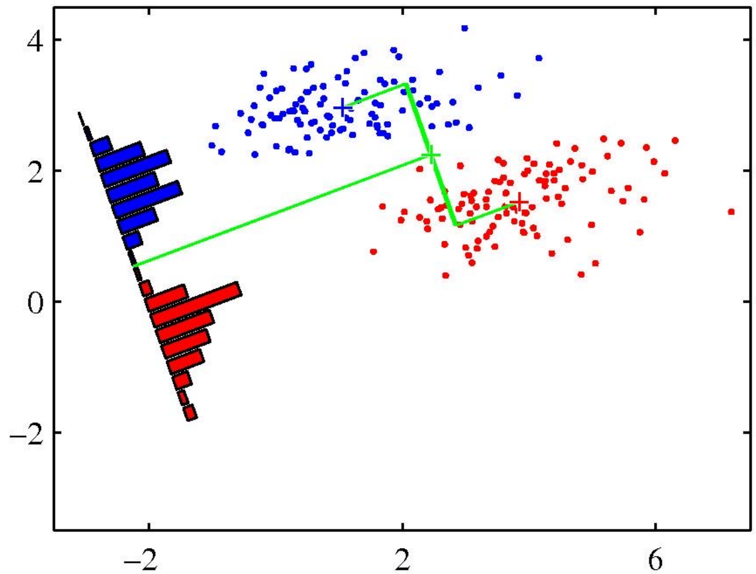
Discriminant : $g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$

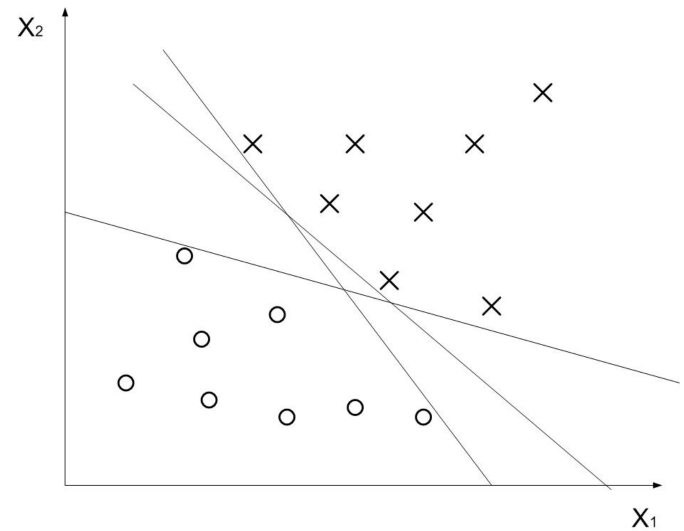
Directly estimate the discrimination function.

Claim: Estimating class densities is harder than estimating the discriminant.

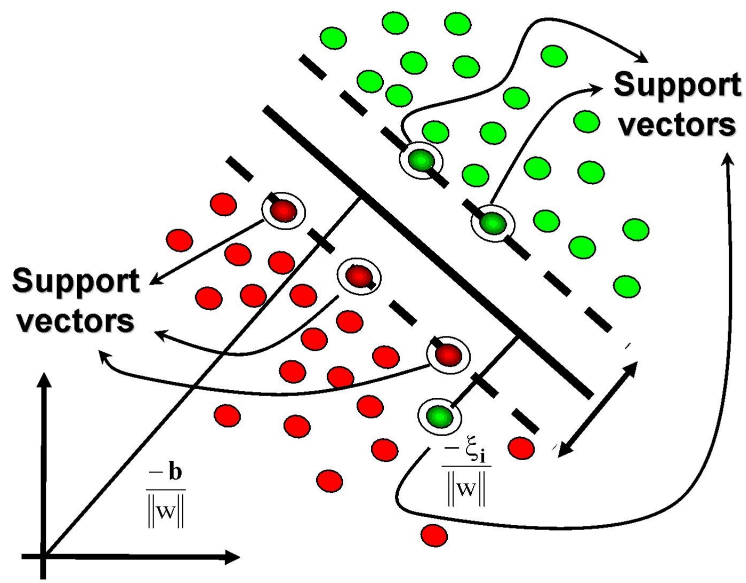
Examples:
Logistic discriminant, LDA, perceptron, SVM...

Linear Discrimination Examples





LDA, perceptron, SVM...



Likelihood vs Discrimination

- Discriminant-based: Assume a model for $g_i(\boldsymbol{x}|\Phi_i)$; no density estimation
- Directly optimize classification accuracy.

- Estimating the boundaries is enough; no need to accurately estimate the densities inside the boundaries

From Discriminants to Posteriors

$$\operatorname{logit}\left(P\left(C_1 \mid \mathbf{x}\right)\right) = \log \frac{P\left(C_1 \mid \mathbf{x}\right)}{1 - P\left(C_1 \mid \mathbf{x}\right)} = \log \frac{P\left(C_1 \mid \mathbf{x}\right)}{P\left(C_2 \mid \mathbf{x}\right)}$$

$$= \log \frac{p\left(\mathbf{x} \mid C_1\right)}{p\left(\mathbf{x} \mid C_2\right)} + \log \frac{P\left(C_1\right)}{P\left(C_2\right)}$$

$$= \log \frac{(2\pi)^{-d/2} |\Sigma|^{-1/2} \exp\left[-(1/2)\left(\mathbf{x} - \mu_1\right)^T \Sigma^{-1} \left(\mathbf{x} - \mu_1\right)\right]}{(2\pi)^{-d/2} |\Sigma|^{-1/2} \exp\left[-(1/2)\left(\mathbf{x} - \mu_2\right)^T \Sigma^{-1} \left(\mathbf{x} - \mu_2\right)\right]} + \log \frac{P\left(C_1\right)}{P\left(C_2\right)}$$

$$= \mathbf{w}^T \mathbf{x} + w_0$$

$$\text{where } \mathbf{w} = \Sigma^{-1}\left(\mu_1 - \mu_2\right) \quad w_0 = -\frac{1}{2}\left(\mu_1 + \mu_2\right)^T \Sigma^{-1}\left(\mu_1 - \mu_2\right) + \log \frac{P\left(C_1\right)}{P\left(C_2\right)}$$

From Discriminants to Posteriors

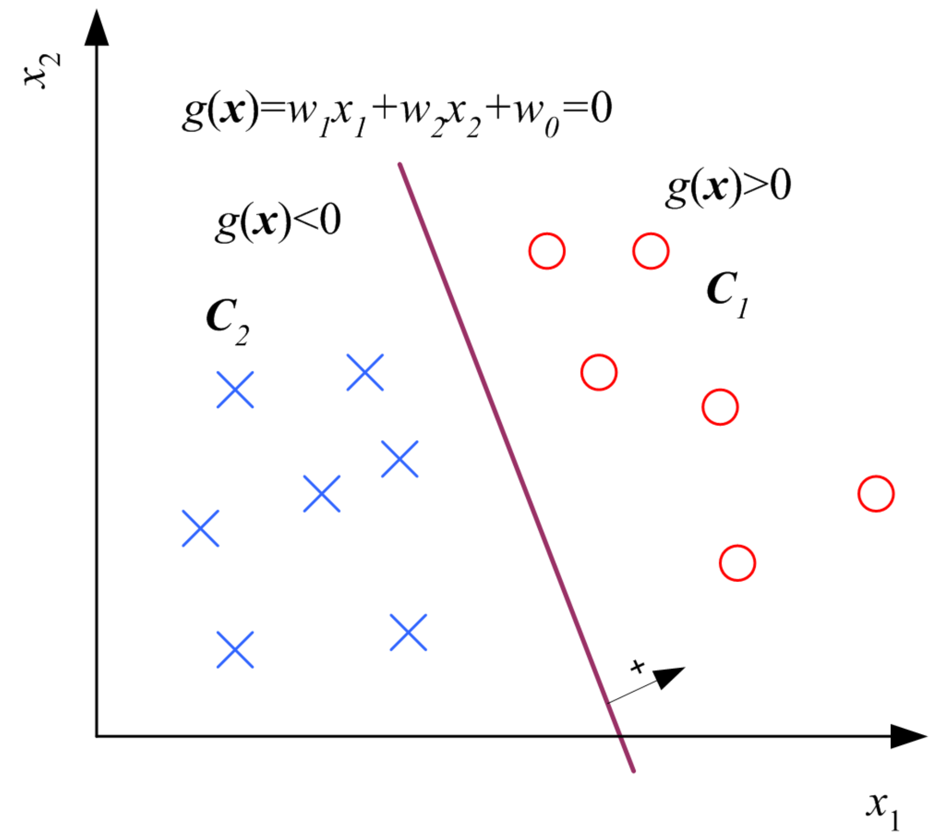
When $p\,(\boldsymbol{x} \mid C_i) \sim N(\boldsymbol{\mu}_i, \Sigma)$, linear discrimination is a log likelihood ratio.

$$y \equiv P(C_1|\mathbf{x}) \text{ and } P(C_2|\mathbf{x}) = 1 - y$$

choose C_1 if $\log\left[y/(1-y)\right] > 0$

C_2 otherwise

$$\log \frac{P(C_1|\mathbf{x})}{1 - P(C_1|\mathbf{x})} = \mathbf{w}^T\mathbf{x} + w_0$$



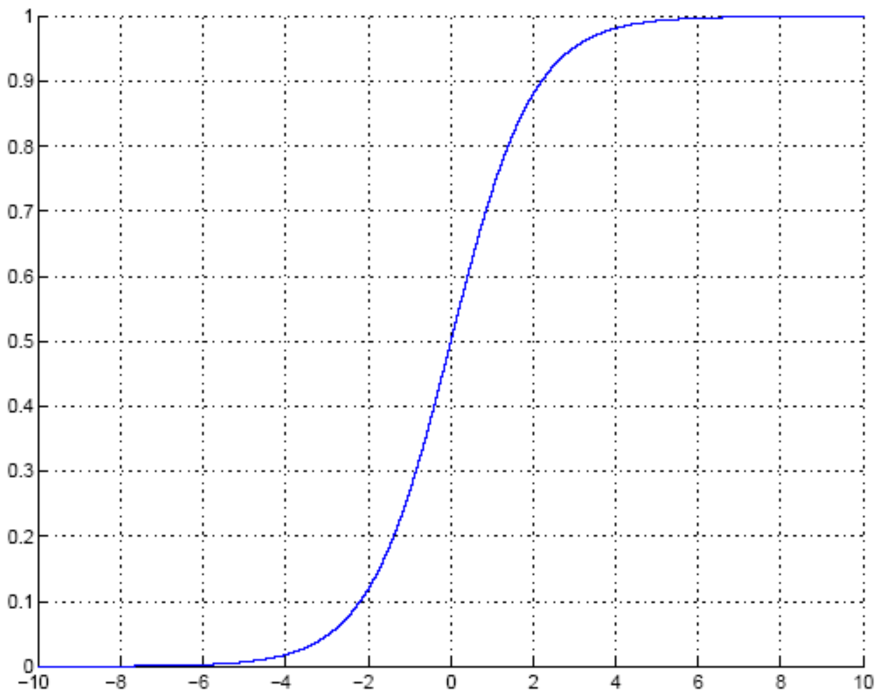
Sigmoid (Logistic) Function

The logit $p(C_1/x)$ is a log odds and the inverse is logistic/sigmoid function.

$$\log \frac{P(C_1|\mathbf{x})}{1-P(C_1|\mathbf{x})} = \mathbf{w}^T\mathbf{x} + w_0$$

$$P(C_1|\mathbf{x}) = \text{sigmoid}\left(\mathbf{w}^T\mathbf{x} + w_0\right) = \frac{1}{1+\exp\left[-\left(\mathbf{w}^T\mathbf{x} + w_0\right)\right]}$$

Sigmoid (Logistic) Function



1. Calculate $g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$ and choose C_1 if $g(\mathbf{x}) > 0$, or

2. Calculate $y = \text{sigmoid}(\mathbf{w}^T\mathbf{x} + w_0)$ and choose C_1 if $y > 0.5$

Training: Two Classes

- Directly minimize negative log Posterior function

$$\mathcal{X} = \{\mathbf{x}^t, r^t\}_t \quad r^t \mid \mathbf{x}^t \sim \text{Bernoulli}(y^t)$$

$$y = P(C_1 \mid \mathbf{x}) = \frac{1}{1 + \exp\left[-\left(\mathbf{w}^T\mathbf{x} + w_0\right)\right]}$$

$$l(\mathbf{w}, w_0 \mid \mathcal{X}) = \prod_t \left(y^t\right)^{\left(r^t\right)}\left(1 - y^t\right)^{\left(1 - r^t\right)}$$

$$E(\mathbf{w}, w_0 \mid \mathcal{X}) = -\log l = -\sum_t r^t \log y^t + \left(1 - r^t\right)\log\left(1 - y^t\right)$$

Gradient-Descent

- E(***w***|X) is error with parameters ***w*** on sample X

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} E(\mathbf{w} \mid X)$$

- Gradient $\nabla_w E = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \dots, \frac{\partial E}{\partial w_d} \right]^T$

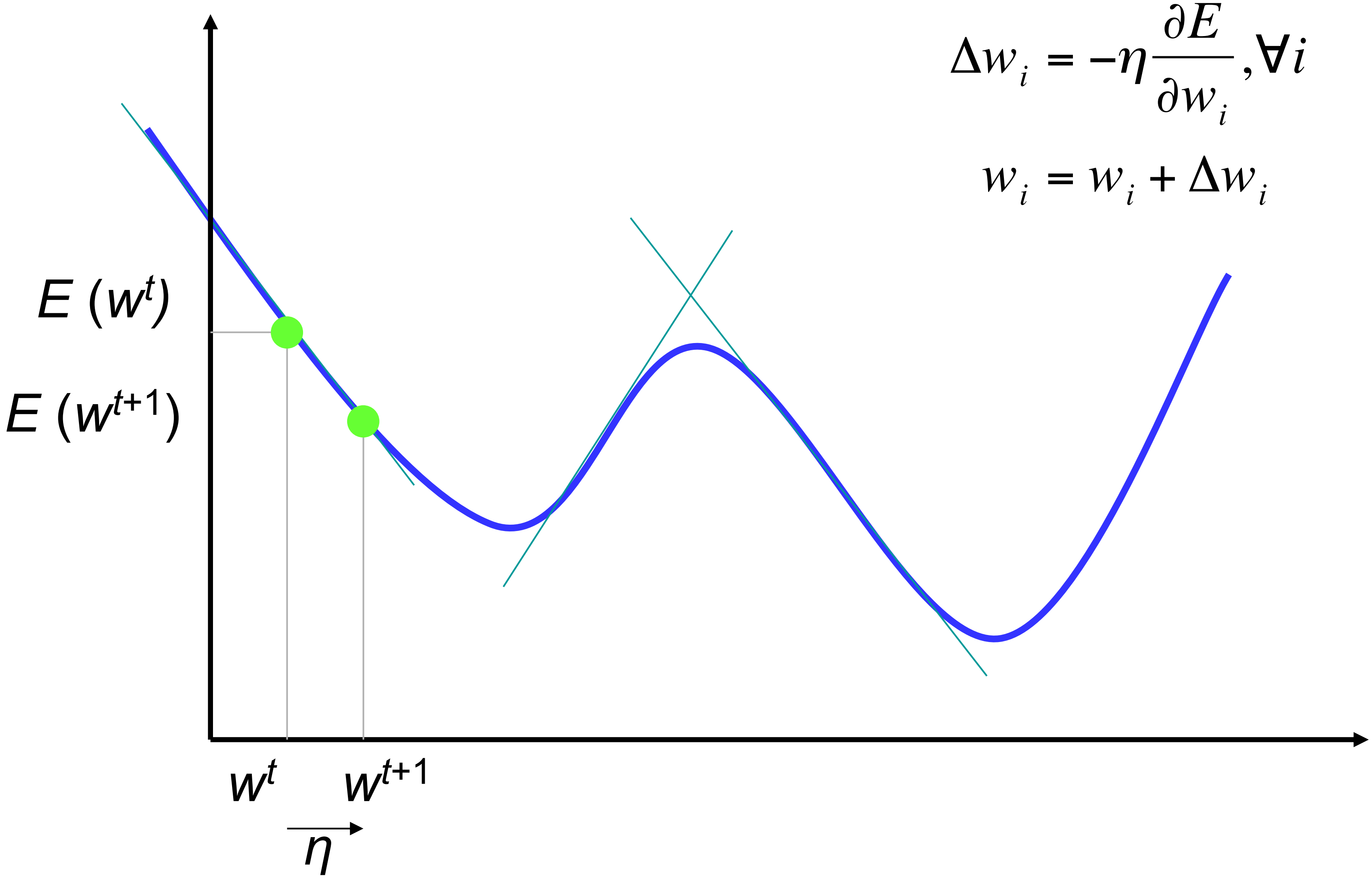
- Gradient-descent:

  Starts from random ***w*** and updates ***w*** iteratively in the negative direction of gradient

Gradient-Descent

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}, \forall i$$

$$w_i = w_i + \Delta w_i$$



Training: Gradient-Descent

$$E(\mathbf{w}, w_0 \mid \mathcal{X}) = -\sum_t r^t \log y^t + (1 - r^t) \log (1 - y^t)$$

$$y = \text{sigmoid}(a) \quad \frac{dy}{da} = y(1 - y)$$

$$a = w^T x + w_0 \quad \frac{da}{dw_j} = x_j$$

$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial a} \frac{\partial a}{\partial w_j} = -\sum_t \left(\frac{r^t}{y^t} - \frac{1 - r^t}{1 - y^t} \right) y^t (1 - y^t) x_j^t$$

$$\frac{\partial E}{\partial w_0} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial a} \frac{\partial a}{\partial w_0} = -\sum_t \left(\frac{r^t}{y^t} - \frac{1 - r^t}{1 - y^t} \right) y^t (1 - y^t)$$

Training: Gradient-Descent

$$E(\mathbf{w}, w_0 \mid \mathcal{X}) = -\sum_t r^t \log y^t + (1 - r^t) \log (1 - y^t)$$

$$\Delta w_j = -\eta \frac{\partial E}{\partial w_j} = \eta \sum_t \left(\frac{r^t}{y^t} - \frac{1 - r^t}{1 - y^t} \right) y^t (1 - y^t) x_j^t$$

$$= \eta \sum_t (r^t - y^t) x_j^t, j = 1, \ldots, d$$

$$\Delta w_0 = -\eta \frac{\partial E}{\partial w_0} = \eta \sum_t (r^t - y^t)$$

```
For j = 0,...,d
    w_j ← rand(-0.01,0.01)
Repeat
    For j = 0,...,d
        Δw_j ← 0
    For t = 1,...,N
        o ← 0
        For j = 0,...,d
            o ← o + w_j x_j^t
        y ← sigmoid(o)
        For j = 0,...,d
            Δw_j ← Δw_j + (r^t − y) x_j^t
    For j = 0,...,d
        w_j ← w_j + η Δw_j
Until convergence
```

Matrix Version of Gradeint Descent

$$w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}, r = \begin{bmatrix} r^1 \\ r^2 \\ \vdots \\ r^N \end{bmatrix}, y = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^N \end{bmatrix}, X = \begin{bmatrix} 1 & x_1^1 & x_2^1 & \cdots & x_d^1 \\ 1 & x_1^2 & x_2^2 & \cdots & x_d^2 \\ \vdots & & & & \\ 1 & x_1^N & x_2^N & \cdots & x_d^N \end{bmatrix}$$

$$\Delta w = \eta X^T \left(r - y \right)$$

$$w^{new} = w^{old} + \eta X^T \left(r - y \right)$$

Perceptron with Error Function

- Perceptron algorithm

$$\text{for each example } (\mathbf{x}^t, r^t):$$
$$if\,(<\mathbf{w}, \mathbf{x}^t > * r^t \leq 0)$$
$$\mathbf{w} = \mathbf{w} + \eta r^t \mathbf{x}^t$$

- One-sample logistic regression

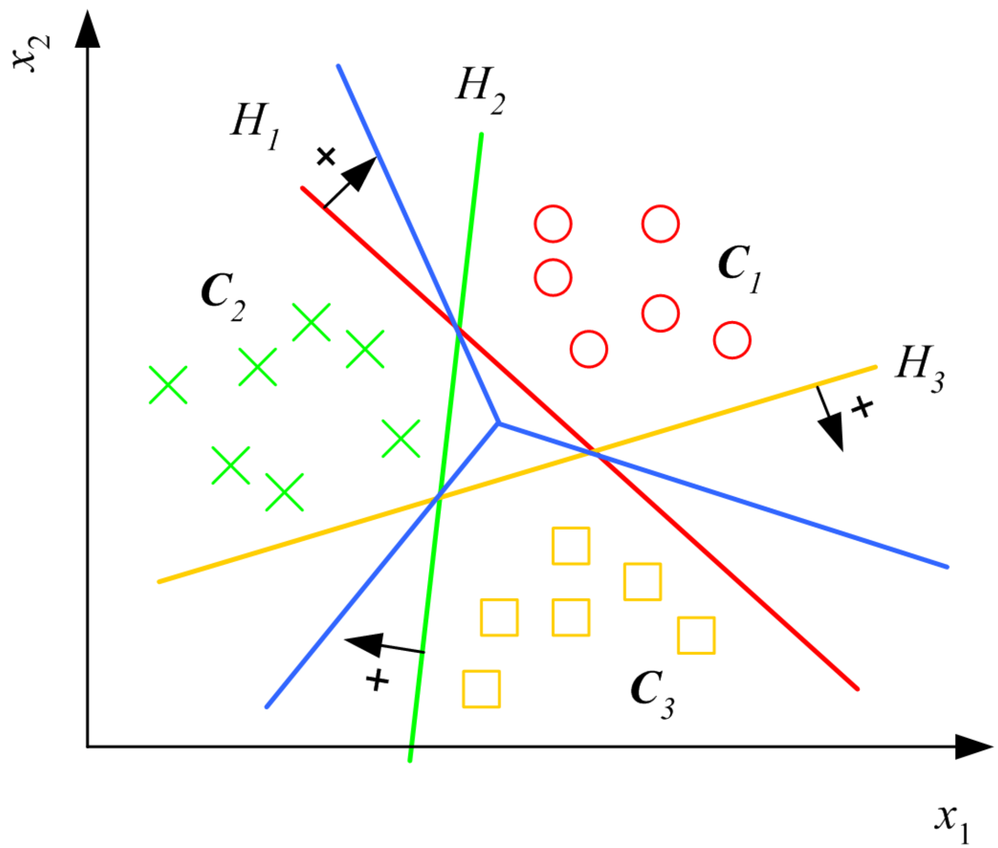
$$y^t = \text{sigmoid}\left(\mathbf{w}^T \mathbf{x}^t\right)$$
$$E^t\left(\mathbf{w}|\mathbf{x}^t, \mathbf{r}^t\right) = -r^t \log y^t - \left(1 - r^t\right) \log \left(1 - y^t\right)$$
$$\mathbf{w} = \mathbf{w} + \eta\left(r^t - y^t\right)\mathbf{x}^t$$

Notes on Logistic Regression

- Optimal when $p(\boldsymbol{x}|C_i)$ are Gaussian with shared cov matrix but no need to worry about cov estimation.
- Useful when classes are (almost) linearly separable.
- The error function is concave with a unique solution with no closed-form solution.
- More efficient algorithms with higher order gradient such as Newton-Raphson (see Bishop's).

Multiple Classes

$$g_i(\mathbf{x} \mid \mathbf{w}_i, w_{i0}) = \mathbf{w}_i^T \mathbf{x} + w_{i0}$$

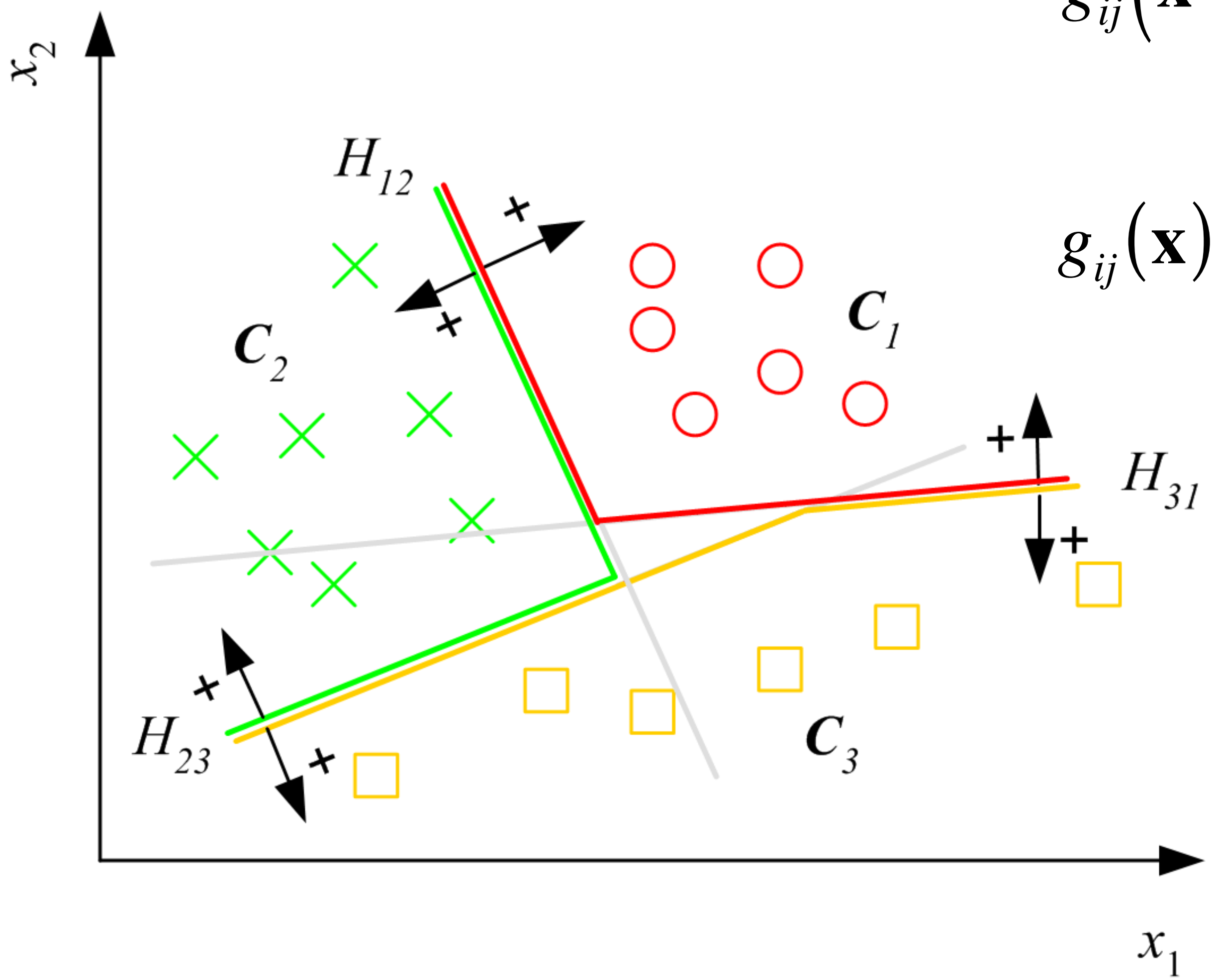


Choose C_i if

$$g_i(\mathbf{x}) = \max_{j=1}^{K} g_j(\mathbf{x})$$

Classes are
linearly separable

Pairwise Separation



$$g_{ij}\left(\mathbf{x} \mid \mathbf{w}_{ij}, w_{ij0}\right) = \mathbf{w}_{ij}^T \mathbf{x} + w_{ij0}$$

$$g_{ij}(\mathbf{x}) = \begin{cases} > 0 & \text{if } \mathbf{x} \in C_i \\ \leq 0 & \text{if } \mathbf{x} \in C_j \\ \text{don't care} & \text{otherwise} \end{cases}$$

choose C_i if

$$\forall j \neq i, g_{ij}(\mathbf{x}) > 0$$

E. Alpaydin, Introduction to Machine Learning

K>2 Classes

- Learn a w for each class;
- Choose a reference class C_K

$$\mathcal{X} = \left\{ \mathbf{x}^t, \mathbf{r}^t \right\}_t \qquad r^t | \mathbf{x}^t \sim \mathrm{Mult}_K \left(1, \mathbf{y}^t \right)$$

$$\log \frac{p(C_i | \mathbf{x})}{p(C_K | \mathbf{x})} = \mathbf{w}_i^T \mathbf{x} + w_{i0}^o$$

$$P(C_K | \mathbf{x}) = \frac{1}{1 + \sum_{j=1}^{K-1} \exp\left[\mathbf{w}_j^T \mathbf{x} + w_{j0} \right]}$$

$$P(C_i | \mathbf{x}) = \frac{\exp\left[\mathbf{w}_i^T \mathbf{x} + w_{i0} \right]}{1 + \sum_{j=1}^{K-1} \exp\left[\mathbf{w}_j^T \mathbf{x} + w_{j0} \right]}, i = 1, \ldots, K\text{-}1$$

$$\boxed{y_i = \hat{P}(C_i | \mathbf{x}) = \frac{\exp\left[\mathbf{w}_i^T \mathbf{x} + w_{i0} \right]}{\sum_{j=1}^{K} \exp\left[\mathbf{w}_j^T \mathbf{x} + w_{j0} \right]}, i = 1, \ldots, K}$$

Softmax: taking the max on a differentiable function.

K>2 Classes

$$y_i = \hat{P}(C_i \mid \mathbf{x}) = \frac{\exp\left[\mathbf{w}_i^T \mathbf{x} + w_{i0}\right]}{\sum_{j=1}^{K} \exp\left[\mathbf{w}_j^T \mathbf{x} + w_{j0}\right]}, i = 1, \ldots, K$$

$$\frac{\partial y_i}{\partial a_j} = y_i(\delta_{ij} - y_j)$$

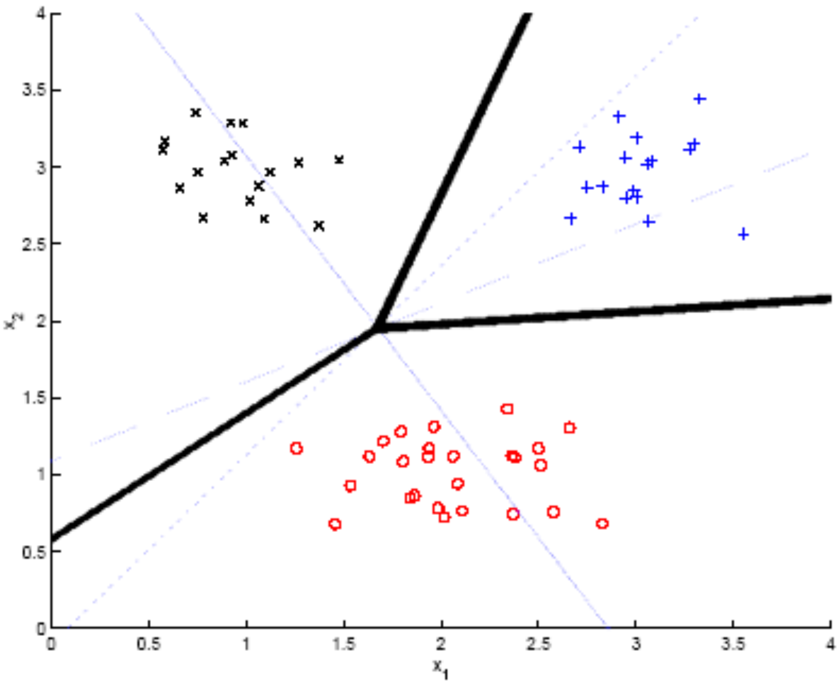
$$l\left(\left\{\mathbf{w}_i, w_{i0}\right\}_i \mid \mathcal{X}\right) = \prod_t \prod_i \left(y_i^t\right)^{\left(r_i^t\right)}$$

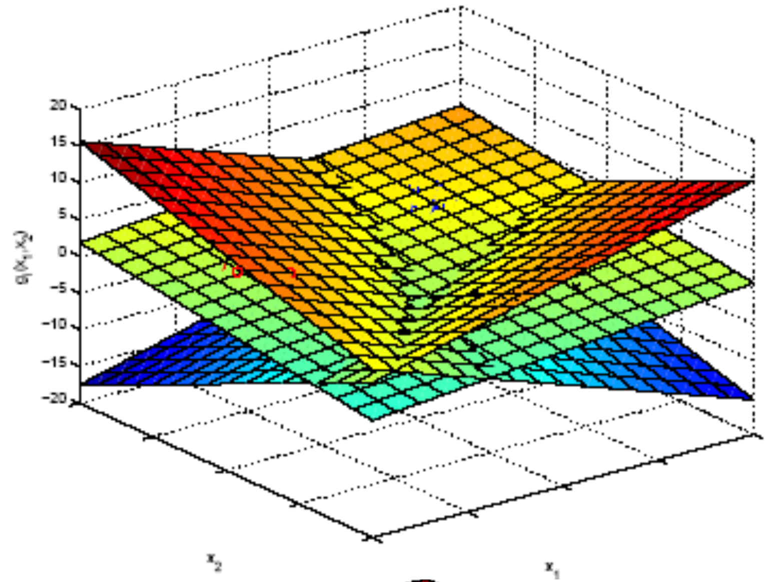
$$E\left(\left\{\mathbf{w}_i, w_{i0}\right\}_i \mid \mathcal{X}\right) = -\sum_t \sum_i r_i^t \log y_i^t$$

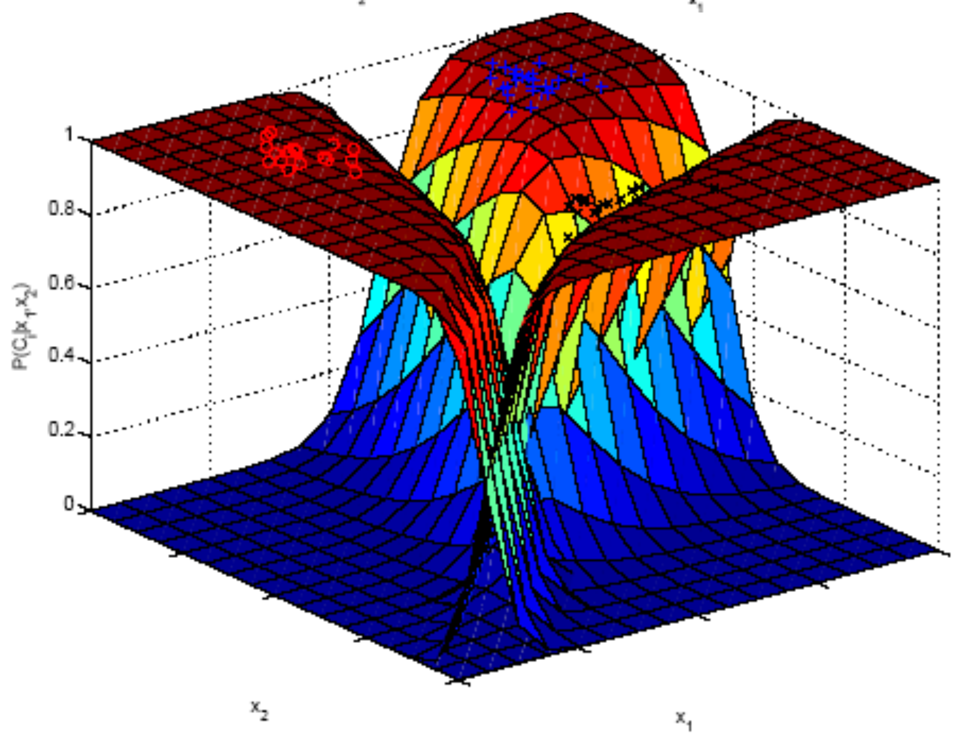
$$\Delta \mathbf{w}_j = \eta \sum_t \sum_i \frac{r_i^t}{y_i^t} y_i^t (\delta_{ij} - y_j^t) \mathbf{x}^t = \eta \sum_t \left(r_j^t - y_j^t\right) \mathbf{x}^t$$

$$\Delta w_{j0} = \eta \sum_t \left(r_j^t - y_j^t\right)$$

Example







E. Alpaydin, Introduction to Machine Learning

Summary of Logistic Regression

- The computational trick:

$$\text{Chain rule}: \quad \frac{\partial E}{\partial w} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial a}\frac{\partial a}{\partial w}$$

Decided by the error function

$$y = \text{sigmoid}(a)$$

$$\frac{dy}{da} = y(1-y)$$

$$a = w^T x + w_0$$

$$\frac{da}{dw} = x$$

$$\text{Iterative update}: \quad w^{new} = w^{old} + \Delta w = w^{old} - \eta \frac{\partial E}{\partial w}$$

Variations

- Other link functions:

$$\text{Chain rule}: \frac{\partial E}{\partial w} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial a}\frac{\partial a}{\partial w}$$

Binomial distribution

$$y = a \quad \frac{dy}{da} = 1$$

$$y = \log(\mathrm{a}) \quad \frac{dy}{da} = \frac{1}{a}$$

$$a = w^T x + w_0$$

$$\frac{da}{dw} = x$$

$$l(\mathbf{w}, w_0 \mid \mathcal{X}) = \prod_t (y^t)^{(r^t)} (1 - y^t)^{(1-r^t)}$$

$$E(\mathbf{w}, w_0 \mid \mathcal{X}) = -\log l = -\sum_t r^t \log y^t + (1 - r^t) \log (1 - y^t)$$