Intro to ML

November 3rd, 2021

Logistics

- 10 min for TA today
- Monday (11/8) TA sessions
- Wednesday 11/10 finish up SVM

- 11/22 midterm covers everything in class up to SVM

CHAPTER 10:

Linear Discrimination

Likelihood- vs. Discriminant-based Classification

- Likelihood-based: Assume a model for $p(\boldsymbol{x}|C_i)$, use Bayes' rule to calculate $P(C_i|\boldsymbol{x})$

  $$g_i(\boldsymbol{x}) = \log P(C_i|\boldsymbol{x})$$

Just any form of equations

- Discriminant-based: Assume a **model** for $g_i(\boldsymbol{x}|\Phi_i)$; not density estimation
- Estimating the boundaries is enough; no need to accurately estimate the densities inside the boundaries
- Inductive bias come from your assumption of boundary not the density itself
- Knowing **how to separate** is more important, learning to separate, not learning to estimate pdfs

Linear Discriminant

- Linear discriminant function (assuming a linear separation):

$$g_i(\mathbf{x} \mid \mathbf{w}_i, w_{i0}) = \mathbf{w}_i^T \mathbf{x} + w_{i0} = \sum_{j=1}^{d} w_{ij} x_j + w_{i0}$$

- Advantages:
  - Simple: O(d) space/computation
  - Knowledge extraction: Weighted sum of attributes; **positive**/**negative** weights (interpretation), magnitudes (importance)
  - Optimal when $p(\boldsymbol{x}|C_i)$ are Gaussian with shared cov matrix; useful when classes are (almost) linearly separable
- Define a error function, find the parameter by gradient decent

Generalized Linear Model

- Quadratic discriminant:

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

Keep discriminant function linear, but transform the input feature

- Adding Higher-order (product) terms:

$$z_1 = x_1,\ z_2 = x_2,\ z_3 = x_1^2,\ z_4 = x_2^2,\ z_5 = x_1 x_2$$

map from ***x*** to ***z*** using nonlinear basis functions and use a linear discriminant in ***z***-space

$$g_i(\mathbf{x}) = \sum_{j=1}^{k} w_{ij} \boxed{\phi_j(\mathbf{x})}$$

Basis function

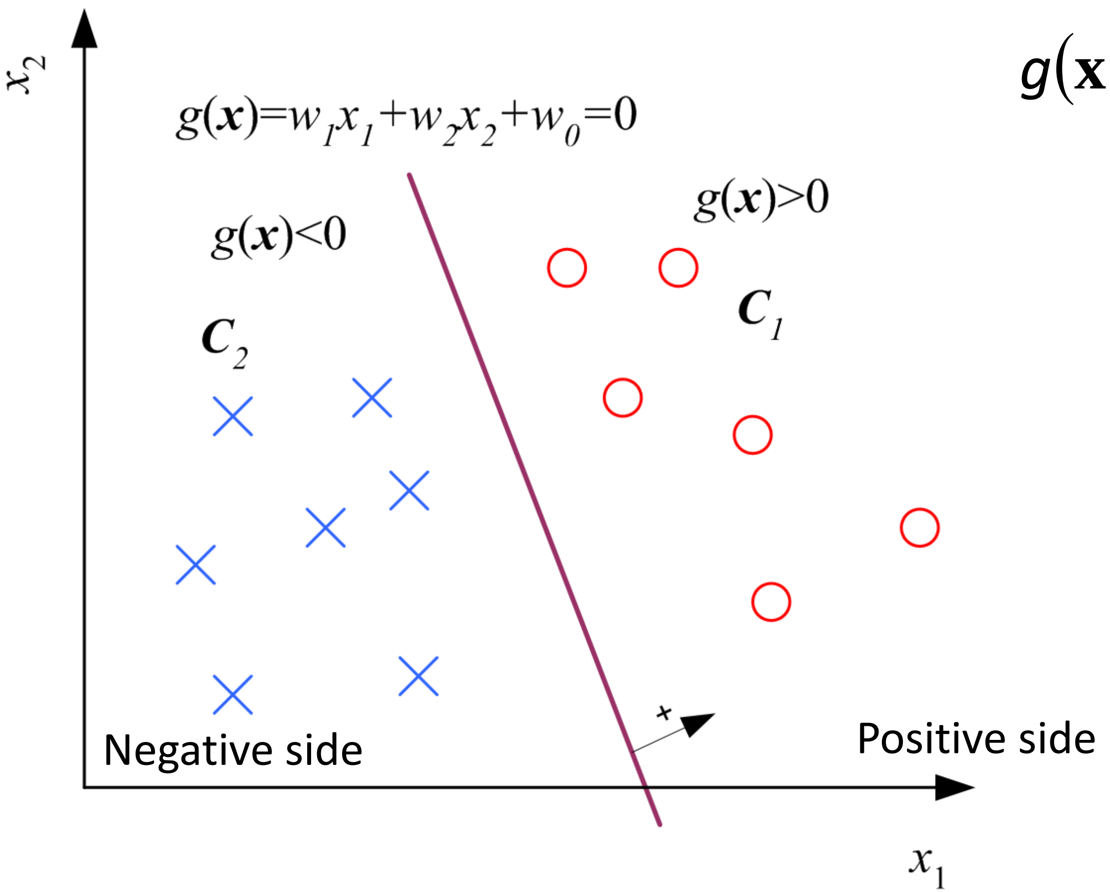
What could be basis function?

- Sin(x1)
- Exp(-(x1-m)^2/c)
- Log(X2)
- 1(x1>c)
- …

Writing nonlinear function as linear function of nonlinear basis -> potential functions (1964)

- This will also be important when we talk about support vector machine with kernel trick

Two Classes



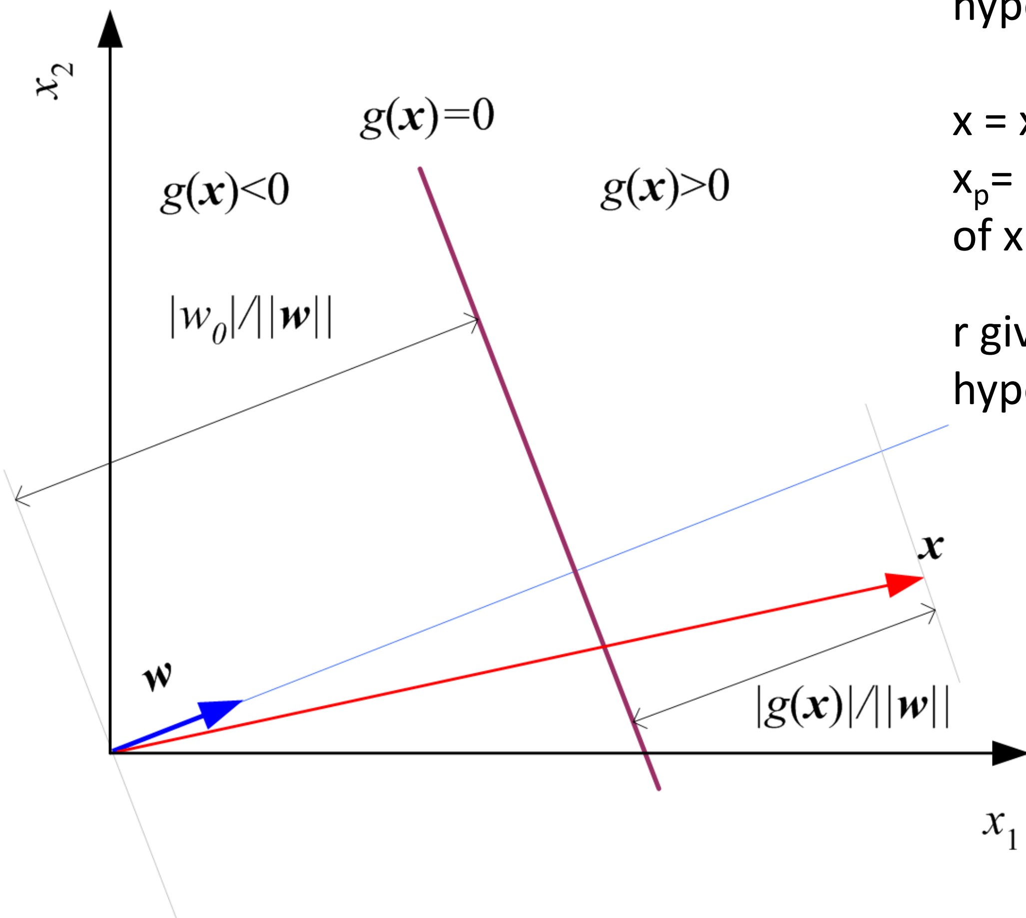
$$
\begin{aligned}
g(\mathbf{x}) &= g_1(\mathbf{x}) - g_2(\mathbf{x}) \\
&= \left(\mathbf{w}_1^T\mathbf{x} + w_{10}\right) - \left(\mathbf{w}_2^T\mathbf{x} + w_{20}\right) \\
&= \left(\mathbf{w}_1 - \mathbf{w}_2\right)^T\mathbf{x} + \left(w_{10} - w_{20}\right) \\
&= \mathbf{w}^T\mathbf{x} + w_0
\end{aligned}
$$

Threshold

Weight vector

$$
\text{choose} \begin{cases} C_1 & \text{if } g(\mathbf{x}) > 0 \\ C_2 & \text{otherwise} \end{cases}
$$

Geometry

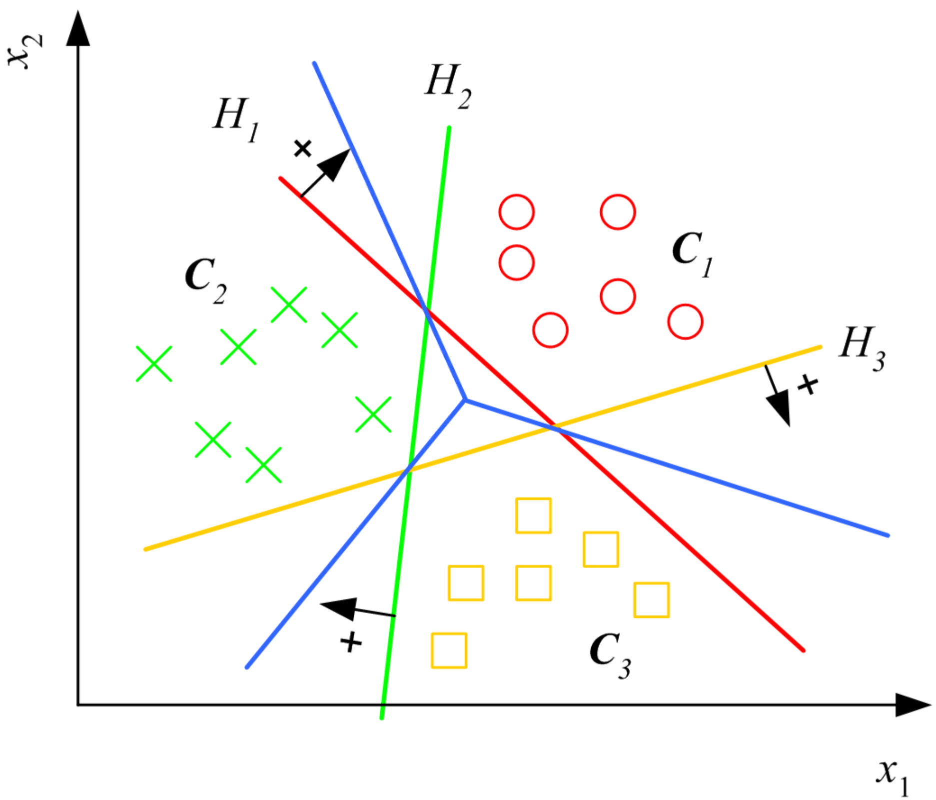


For x1 x2 on decision plane
$w^T(x1-x2) = 0$
w is normal to any vector lying on the decision hyperplane

$x = x_p + r\ w/||w||$
x_p= is the normal projection of x onto hyperplane

r gives us distance from x to hyperplane

Multiple Classes



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

Classes are
linearly separable

g(i) positive for only one class

Choose C_i if

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

From Discriminants to Posteriors

When $p\ (\boldsymbol{x} \mid C_i) \sim \mathrm{N}\ (\boldsymbol{\mu}_i\ , \Sigma)$

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

$$\mathbf{w}_i = \Sigma^{-1}\mu_i \quad w_{i0} = -\frac{1}{2}\mu_i^T \Sigma^{-1}\mu_i + \log P\left(C_i\right)$$

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

$$\text{choose } C_1 \text{ if} \begin{cases} y > 0.5 \\ y/(1-y) > 1 \\ \log\left[y/(1-y)\right] > 0 \end{cases} \text{ and } C_2 \text{ otherwise}$$

Log(y/(1-y))-> logit transform
Log odds of y

$$\text{logit}(P(C_1|\mathbf{x})) = \log\frac{P(C_1|\mathbf{x})}{1-P(C_1|\mathbf{x})} = \log\frac{P(C_1|\mathbf{x})}{P(C_2|\mathbf{x})}$$

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

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

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

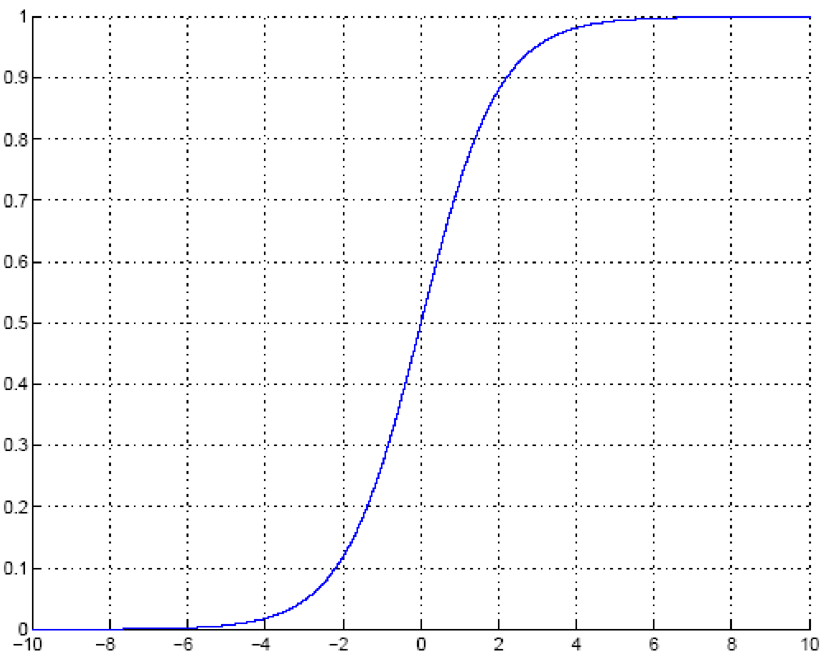
$$\text{where } \mathbf{w} = \Sigma^{-1}(\mu_1-\mu_2) \quad w_0 = -\frac{1}{2}(\mu_1+\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2)$$

The inverse of logit ← Logistic function

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

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

Sigmoid (Logistic) Function



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

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

Sigmoid(0)=0.5, this function takes discriminant function to posterior probability

Gradient-Descent

- E($\boldsymbol{w}$|X) is error with parameters $\boldsymbol{w}$ on sample X

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

Many do not have analytical solution

- Gradient

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

- Gradient-descent:

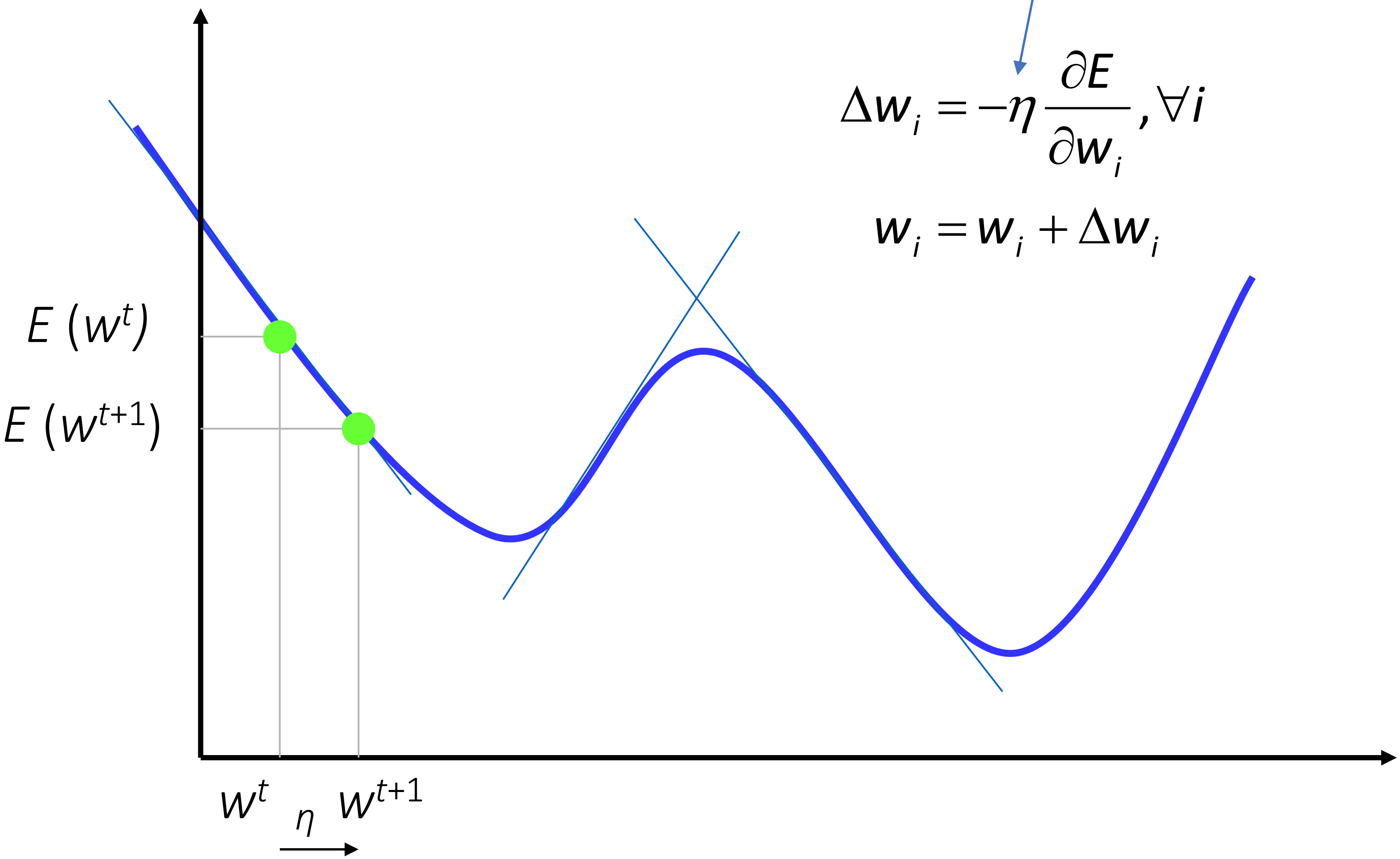
  Starts from random $\boldsymbol{w}$ and updates $\boldsymbol{w}$ iteratively in the negative direction of gradient

Gradient-Descent

Step size

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

$$w_i = w_i + \Delta w_i$$



Logistic Discrimination

Two classes: Assume log likelihood ratio is linear

$$\log \frac{p(\mathbf{x}|C_1)}{p(\mathbf{x}|C_2)} = \mathbf{w}^T\mathbf{x} + w_0^o$$

$$\text{logit}(P(C_1|\mathbf{x})) = \log \frac{P(C_1|\mathbf{x})}{1-P(C_1|\mathbf{x})} = \log \frac{p(\mathbf{x}|C_1)}{p(\mathbf{x}|C_2)} + \log \frac{P(C_1)}{P(C_2)}$$

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

$$\text{where } w_0 = w_0^o + \log \frac{P(C_1)}{P(C_2)}$$

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

Training: Two Classes

Model label given x with probability y

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

Note the difference to likelihood method

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

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

Maximize this function based on data we have

$$E = -\log l$$

Minimize this

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

What is this? This is a function that we call ‘cross entropy’

Training: Gradient-Descent

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

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

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

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

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

Good practice: Z-normalize features

Keep is close to zero

For $j = 0, \ldots, d$

$w_j \leftarrow$rand(-0.01,0.01)

Repeat

For $j = 0, \ldots, d$

$\Delta w_j \leftarrow 0$

For $t = 1, \ldots, N$

$o \leftarrow 0$

For $j = 0, \ldots, d$

$o \leftarrow o + w_j x_j^t$

$y \leftarrow \text{sigmoid}(o)$

$\Delta w_j \leftarrow \Delta w_j + (r^t - y) x_j^t$

For $j = 0, \ldots, d$

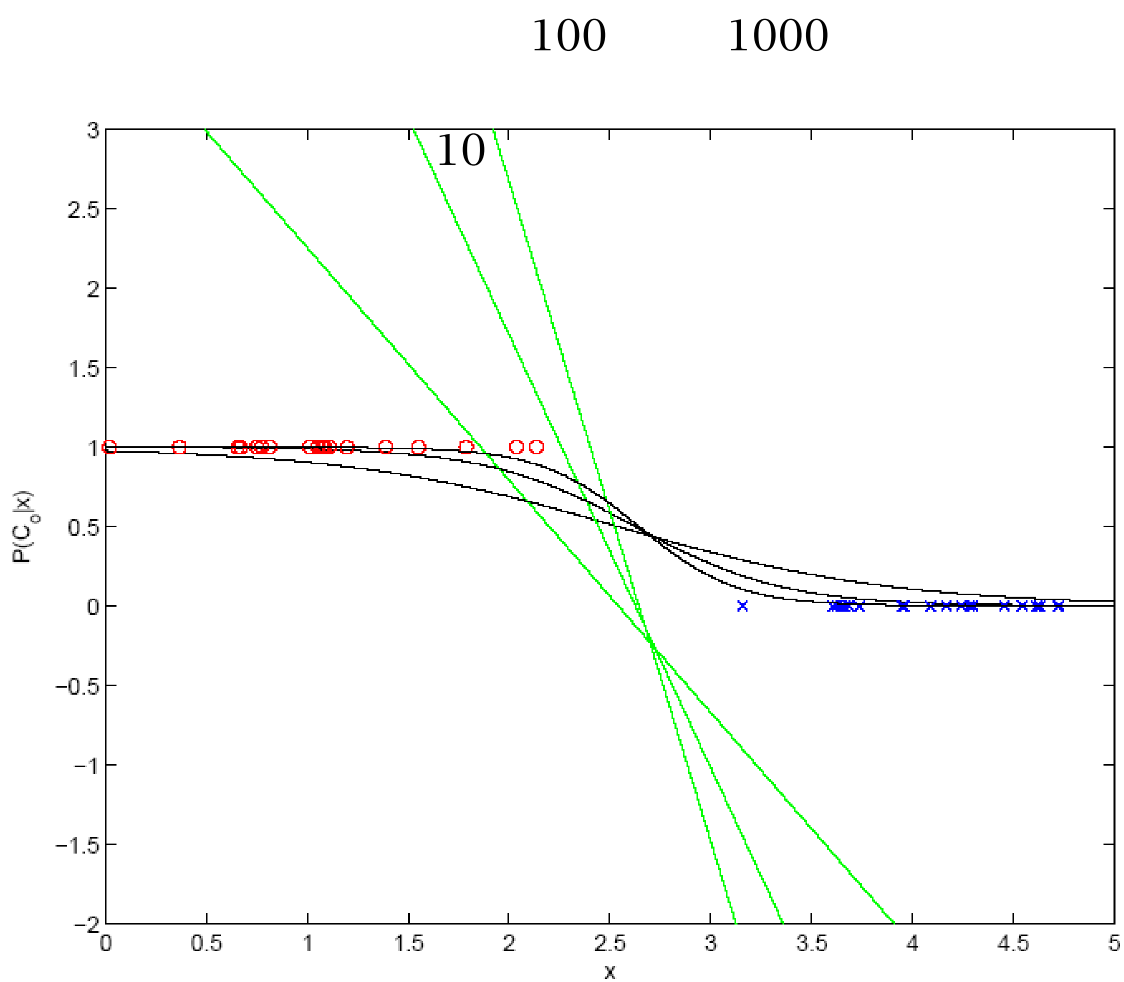
$w_j \leftarrow w_j + \eta \Delta w_j$

Until convergence

Notes

- Gradient does not change anymore then converged
- In this case, we assume log ratio of class density is linear to perform this learning (but we never explicitly estimate $p(x|Ci)$ or $P(Ci)$
- Training effectively takes data of a class to result in either $y<0.5$ or $y>0.5$

1d Example



Keep iteration without stopping, make the sigmoid function harden (quickly takes the sample to close to 0 or 1)

But does not change misclassification rate

Early stoping

K>2 Classes

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

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

C_k is the reference class

Assume linear form

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

wi0= w_{i0}^o + logP(Ci)/P(Ck)

$$\sum_{i=1}^{K} \frac{P(C_i|x)}{P(C_k|x)} = \frac{1 - P(C_k|x)}{P(C_k|x)} = \sum_{i=1}^{K-1} \exp[w_i^T x + w_{i0}]$$

$$P(C_k|x) = \frac{1}{1 + \sum_{i=1}^{K-1} \exp[w_i^T x + w_{i0}]}$$

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

Softmax

It's like a max function

Only its differentiable

Learning the paramter

rt|xt
Multinomial
distribution

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

$$E(\{\mathbf{w}_i, w_{i0}\}_i|\mathrm{X}) = -\sum_t r_i^t \log\, y_i^t$$

$$\Delta\mathbf{w}_j = \eta \sum_t \left(r_j^t - y_j^t\right)\mathbf{x}^t \quad \Delta w_{j0} = \eta \sum_t \left(r_j^t - y_j^t\right)$$

Generalizing the Linear Model

- Quadratic:

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

- Sum of basis functions:

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

where $\phi(x)$ are basis functions. Examples:

  - Hidden units in neural networks (Chapters 11, 12, 13)
  - Kernels in SVM (Chapter 14)