L6

Naive Bayes

$P(X|y) = \prod_{j=1}^{n} P(x_j|y)$

$P(y=1) = \phi_y$

$P(x_j=1|y=0) = \phi_{j|y=0}$

$P(x_j=0|y=1) = \phi_{j|y=1}$

MLE $\phi_y = \Sigma 1\{y^{(i)}=1\}$

$\phi_{j|y=0} = \dfrac{\sum^m 1\{x_j^{(i)}=1, y^{(i)}=0\}}{\sum^m 1\{y^{(i)}=0\}}$

At prediction time

$P(y=1|x) = \dfrac{p(x|y=1)\,p(y=1)}{p(x|y=1)p(y=1) + p(x|y=0)\,p(y=0)}$ (eq)

When it breaks down

say sth is 0 prob just because you haven't 0

eg "NIPS" in spam email

$j = 6017$ $P(x_{6017}=1|y=1) = \frac{0}{\#\{y=1\}} = \phi_{6017|y=1}$

$P(x_{6017}=1|y=0) = \frac{0}{\#\{y=0\}} = \phi_{6017|y=0}$

going back to equation (eq)

$P(y=1|x) = \frac{0}{0+0}$ → very bad

Another example

Stanford football team all lost

x win or lost

$p(x=1) = 0$

Very bad estimate → Laplace smooth $\frac{\#\text{"1"}+1}{\#\text{"0"}+1}$

$p(x=1) = \frac{1}{6}$, much more possible

why? uniform Bayes prior, optimal, see advanced Bayes courses

After Laplace smoothing ~~Estimate~~

$P(x=j) = \dfrac{\Sigma\{x^{(i)}=j\}+1}{M+k}$

$\phi_{j|y=0} = \dfrac{\sum^m \{x_j=1, y=0\}+1}{\sum^m \{\{y^{(i)}=0\}\}+2}$

What if $x_i \in \{1 - k\}$

discretization

	400-800	800-1200
x_2	1	2

$P(x|y) = \prod_{j=1}^{n} P(x_j|y)$ multinomial

Multinomial Bernoulli model

$x \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ ← a ← advark

Disadvantage of naive bayes if same word appear twice, feature vector doesn't show this

Another representation Multinomial Event model

$x \in \begin{bmatrix} 1600 \\ 800 \\ 1600 \\ 6200 \end{bmatrix} \in \mathbb{R}^{n_i}$

n_i length of ith email

$P(x,y) = \prod_{j=1}^{n} p(x_j|y)\,p(y) \overset{\text{assume}}{=} p(x|y)\,p(y)$

Parameters $\phi_y = p(y=1)$

$\phi_{k|y=0} = P(x_j = k | y=0)$

Assumption word doesn't depend on position

MLE $\phi_{k|y=0} = \dfrac{\sum^m 1\{y^{(i)}=0\}\left(\sum^{n_i} \{x_j^{(i)}=k\}\right) + 1}{\sum_{i=1}^{m} \{y=0\}\, n_i + 10{,}000}$

Laplace smoothing

notes encounter unknown word ① ignore ② UNK

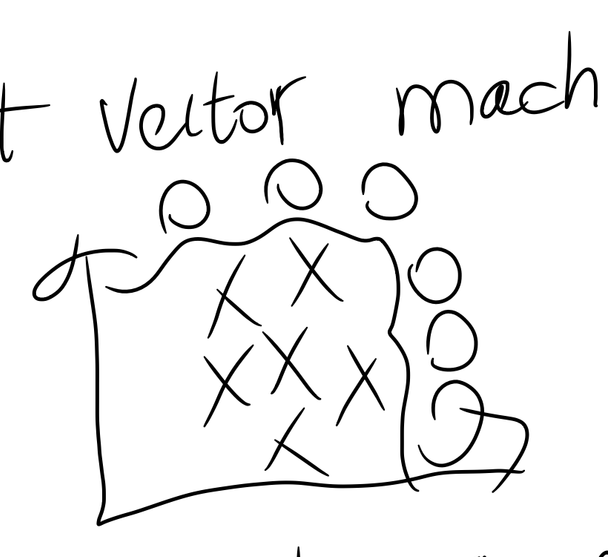
Naive Bayes ① efficient ② underperform logistics

word embedding

Quick and dirty

↳ help classify difficulty of each problem

Support Vector machine



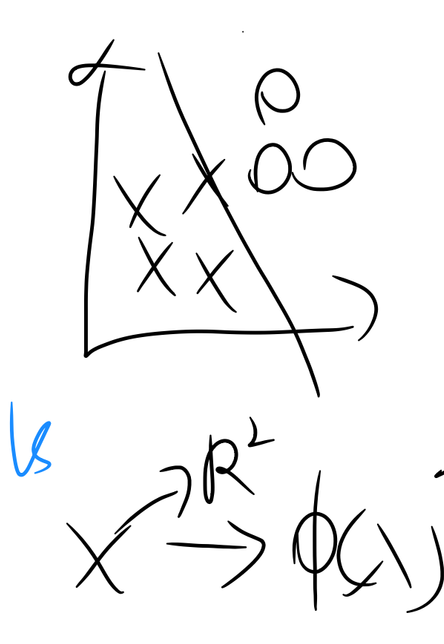
Non-linear decision boundaries

logistics regression can do this by using high-dimensional features vector are linear

eg $\begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \\ x_2^2 \end{bmatrix}$

but choosing is way too painful

— Optimal margin classifier

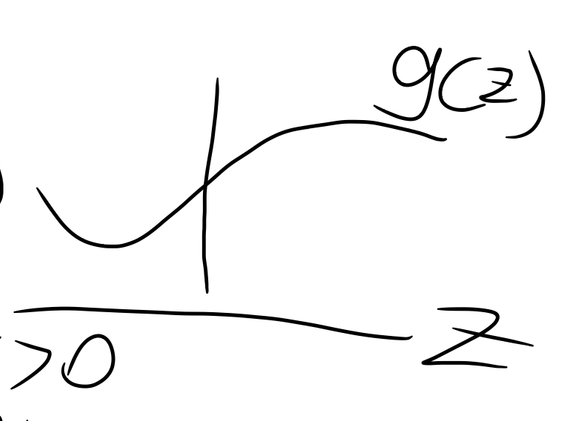


— kernels $x \to \phi(x)$: $\mathbb{R}^2 \to \mathbb{R}^{1000}$ maybe $\mathbb{R}^{\infty}$

— Inseperable case

Functional margin

$h_\theta(x) = g(\theta^T x)$



predict "1" if $\theta^T x > 0$

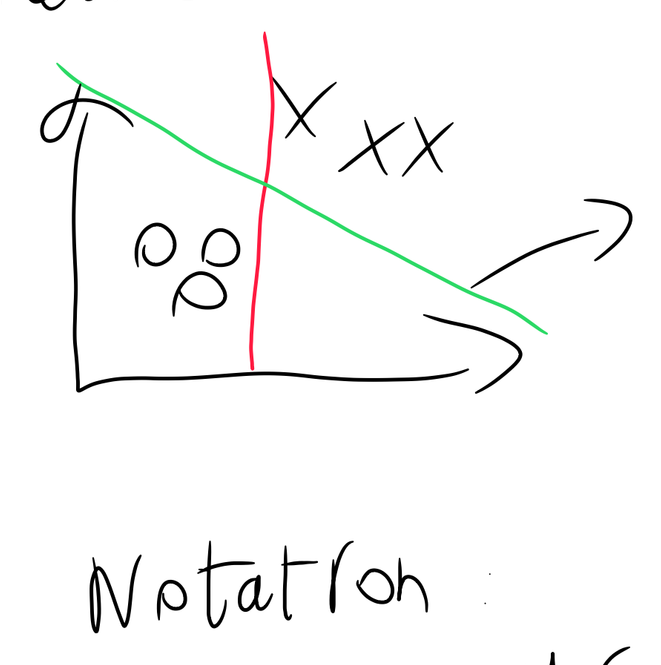
predict "0" if otherwise

If $y^{(i)} = 1$, hope that $\theta^T x^{(i)} \gg 0$

If $y^{(i)} = -1$, hope that $\theta^T x^{(i)} \ll 0$

} want these two statement true (hope statement)

Geometric margin



→ better or greater geometric margin as the line doesn't get close to points

Notation

$y \in \{-1, +1\}$

$g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{otherwise} \end{cases}$

$h_{w,b}(x) = g(w^T x + b)$, $w \in \mathbb{R}^n$, $b \in \mathbb{R}$ Drop $x_0 = 1$ constraint

Functional margin of hyperplane defined by (w,b) with $(x^{(i)}, y^{(i)})$

$\hat{\gamma}^{(i)} = y^{(i)}(w^T x^{(i)} + b)$

with hopestatement

Hope $\hat{\gamma}^{(i)} \gg 0$

If $\hat{\gamma}^{(i)} > 0$ that means $h(x^{(i)}) = y^{(i)}$

Functional margin with respect to whole train set — assume training set is linearly seperable

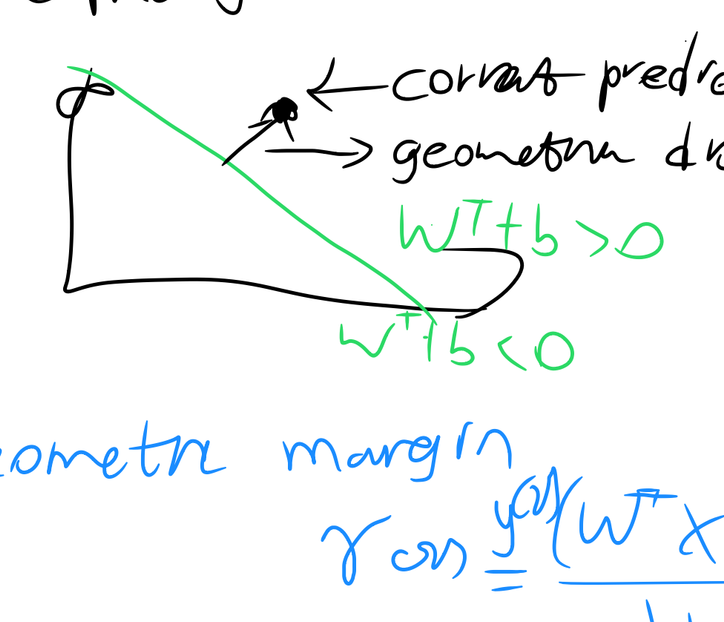
$\hat{\gamma} = \min_i \hat{\gamma}^{(i)}$

functional margin cheatable eg $w \leftrightarrow 2w$, $b \leftrightarrow 2b$ but not really useful

So want to normalize $\|w\| = 1$

by $(w,b) = \left(\frac{w}{\|w\|}, \frac{b}{\|b\|}\right)$

Geometric margin



Geometric margin

$\gamma^{(i)} = \dfrac{y^{(i)}(w^T x^{(i)} + b)}{\|w\|}$

$= \dfrac{\hat{\gamma}^{(i)}}{\|w\|}$

Geometric margin with respect to training set

$\gamma = \min_i \gamma^{(i)}$

Optimal margin classifier

(w,b) to maximize γ

$\Longleftrightarrow \begin{cases} \min_{w,b} \|w\|^2 \\ \text{s.t. } y^{(i)}(w^T x^{(i)} + b) \geq 1 \end{cases}$