Lecture-6

⊛ At prediction time (Ø Naive Bayes):

⊛ $$P(y=1 \mid x) = \frac{P(X \mid y=1) \cdot P(y=1)}{P(X \mid y=1) \cdot P(y=1) + P(X \mid y=0) \cdot P(y=0)}$$

⊛ Lets say there is a word NIPS (j > 6017) in your dictionary but it does not appear in your email:

$$P(X_{6017}=1 \mid y=1) = \frac{0}{\#\{y=1\}}$$

→ Statistically it is a bad idea to say that something will not definately happen.

This mean that during prediction:

$$P(y|x) = \boxed{\prod_{i=1}^{10000} P(x_j|y)} \cdot P(y)$$

$$\boxed{\prod_{i=1}^{10000} P(x_j|y=1) \cdot P(y=1)} + \boxed{\prod_{i=1}^{10000} P(x_j|y=0) \cdot P(y=0)}$$

→ Due to non-presence of ith word - we have not seen before in our dataset → the product highlighted in red will become zero during predict. ith word might be in your testing instance but during training it was not present so product becomes zero.

How to Improve this error of zero?
→ Laplace smoothing:

Example (Stanford football team performance)

		won
9/12	Wake Forest	0
10/17	Arizona	0
11/21	Caltech	0
12/31	Oklahoma	?

→ What should be there winning chance in fourth game?

→ $\dfrac{\#\text{of Wins}}{\#\text{of Wins} + \#\text{of Lost}} = \dfrac{0}{0+3} = 0$ (Absolute certainty of Losing).

→ Again statistically it is bad to assume that with absolute certainity Standford won't win next games

→ Here comes in $\boxed{\text{Laplace smoothing}}$ which means:-

→ Add One to both ~~numerator~~ and ~~one to denominator~~ events.

$$\frac{\#\text{Wins} + \boxed{1}}{\#\text{Wins} \boxed{+1} + \#\text{ of Loses} \boxed{+1}} = \frac{1}{6} \text{ (More reasonable).}$$

* $\boxed{\text{More Generally}}$

$$X \in \{1, \dots k\}$$

Estimate:- (Laplace Smoothing)

$$P(X = j) = \frac{\sum_{j=1}^{m} d\{x^{(i)} = j\} + 1}{m + 1}$$

* Laplace Smoothed Naive Bayes

(a) $$\phi_{y} = P(y) \frac{\sum_{i=1}^{m} d\{y^{(i)} = 1\} \cancel{+1}}{m \; \cancel{+}}$$

(b) $$\phi_{j|y=i} \frac{\sum_{i=1}^{m} \{X_j = 1, y^{(i)} = 1\} +1}{\sum_{i=1}^{m} \{y^{(i)} = 1\} + 2}$$

→ Laplace Smoothed (Do Similar for $\phi_{j|y=0}$).

my)

* What to do when features are multinomial?

$X_i \in \{1, \ldots, k\}$

Size	<400 feet	400< <800	800-1200	>1200
X_i	1	2	3	4

→ Made Buckets of this feature.

$$P(X|y) = \underbrace{\prod_{j=1}^{m} P(X_j|y)}$$

→ Multinomial mean that each feature X_j can now assume more than one value rather than only binary value.

→ $\boxed{\text{multinomial}}$ (More than one classes).

→ Better Representation of Naive Bayes.

(*) So Far = (Multivariate Bernoulli event model)

$$X = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \\ 0 \end{bmatrix} \begin{array}{l} a \leftarrow 1 \\ aardvark \leftarrow 2 \\ \vdots \\ buy \leftarrow 800 \\ \vdots \\ drugs \leftarrow 1600 \\ \vdots \\ now \leftarrow 6200 \\ \end{array} \rightarrow X_i \in \{0,1\}$$

↳ Disregard the count of word

"Drugs! Buy drugs now!"

→ New representation: (Multinomial Event Model)

$$X = \begin{bmatrix} 1600 \\ 800 \\ 1600 \\ 6200 \end{bmatrix} \in \mathbb{R}^n$$ (where n is the length of the sentence, in this case four for the sentence "Drugs! Buy drugs now!"

but $X_j \in \{1, \dots 10000\}$.

different

n_i = length of email i. (Varies for each instance).

(*)

⊛ Lets build a generative model for Multinomial Event model:-

$$P(x, y) = P(x|y) \cdot P(y)$$

assume

$$= \prod_{j=1}^{n} \underbrace{P(x_j|y)}_{\text{Multinomial}} \cdot P(y)$$

n → Depends on single instance (will vary).

→ Parameters

$$\boxed{\phi_y = P(y=1)}, \boxed{\phi_{k|y=0}} = \underbrace{P(x_j = k \mid y=0)}$$

⊛ Chance of a word ~~[illegible]~~ j being ~~k~~ k, if $y = 0$

→ $\phi_{k|y=1} = P(x_j = k \mid y = 1)$.

→ MLE:- (Parameters)

Of all the emails how many times word k appears

$$\phi_{k|y=0} = \frac{\sum_{i=1}^{m} \mathbb{1}\{y^{(i)}=0\} \sum_{j=1}^{n_i} \mathbb{1}\{x_j^{(i)} = k\} + \boxed{1}}{\boxed{\sum_{i=1}^{m} \mathbb{1}\{y^{(i)}=0\} \cdot n_{(i)}^{(i)}} + \boxed{10000}}$$

Total number of words in non-spam emails

Naive Bayes is quick to implement and is computationally efficient.

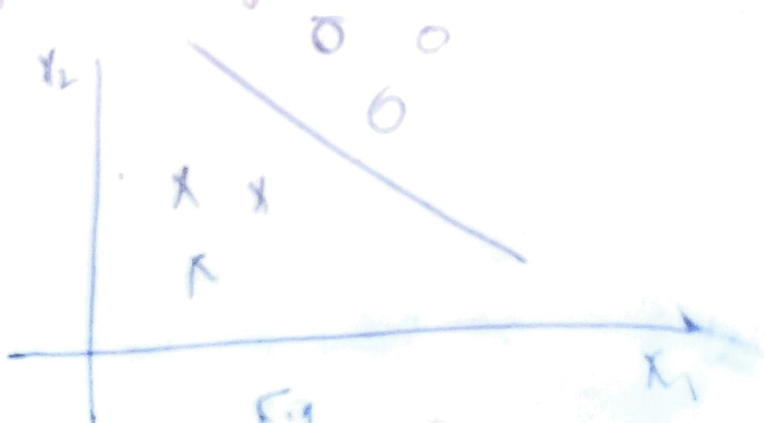
Gaussian and Naive Bayes are quick to implement

Support Vector Machines

Fig. 1

→ Turn Key Property → Support Vector Machine do not have many hyperparameter

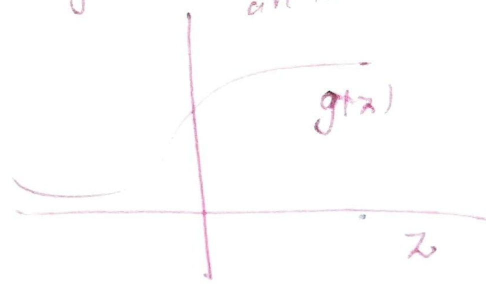
→ Optimal margin classifier (separable case)



Fig

• Functional Margin:- (How confident you are about an instance)

Logistic Regression as motivation example ← $h_\theta(x) = g(\theta^T x)$



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

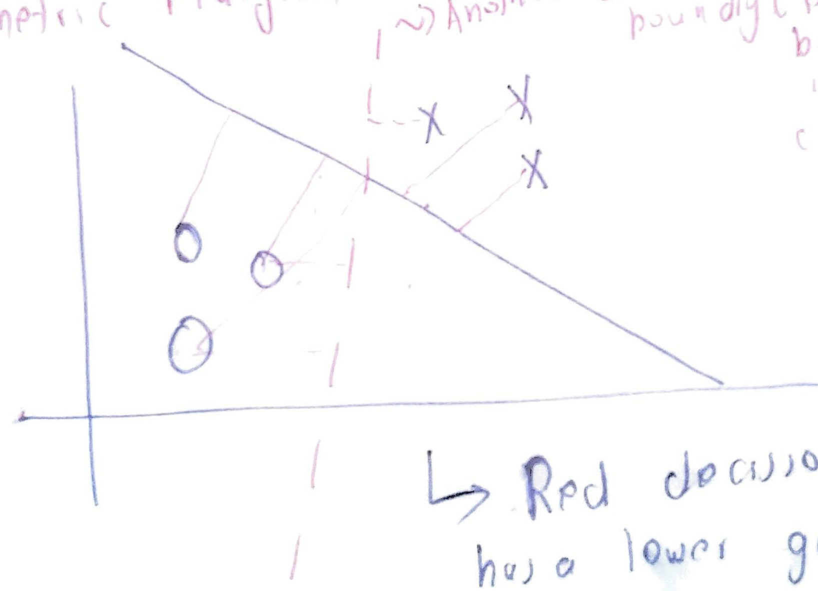
"0" otherwise

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

• if $y^{(i)} = 0$, hope that $\theta^T x^{(i)} \ll 0$ → Much less

• Geometric Margin:

~> Another decision boundry (But is bad because it come close to other examples).



↳ Red decision boundry has a lower geometric

↳ Blue decision boundry (Better!!!) has a higher decision boundry

(ii) Some changes in SVM

Notation:

→ Labels → $y \{-1, +1\}$ → $\boxed{\text{Not } 0,1}$

Have h output values in $\{-1, +1\}$ → $\boxed{\text{Not probability}}$

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

◦ Previous

$h_\theta(x) = g(\theta^T x)$ ← $\mathbb{R}^{n+1}$, $x_0 = 1$.

for SVM → Please not the differences

$h_{w,b}(x) = g(\underbrace{w^T x + b}_{\sum_{i=1}^{n} w_i x_i + b})$ ← $\mathbb{R}^n$, $\mathbb{R}$

* Functional margin of (w, b) $\longrightarrow$ hyperplane
  * Hyperplane defined by (w, b) w.r.t $(x^{(i)}, y^{(i)})$

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

If $y^{(i)} = 1$, want $w^T x^{(i)} + b >> 0$

If $y^{(i)} = -1$, want $w^T x^{(i)} + b << 0$

$\hat{\gamma}^{(i)} >> 0$ (Combining these two)

(Want such a functional margin).

* if $\hat{\gamma}^{(i)} > 0$

that $h(x^{(i)}) = y^{(i)}$

$\rightarrow$ Functional Margin w.r.t traing set:

$$\hat{\gamma} = \min_{i=1 \ldots m} \hat{\gamma}^{(i)}$$

(Worst-case Notion)

You can cheat this defination of functional margin by increasing the ~~w~~ w & b magnitude indefinately (Scaling up) → Would not change the decision boundry.

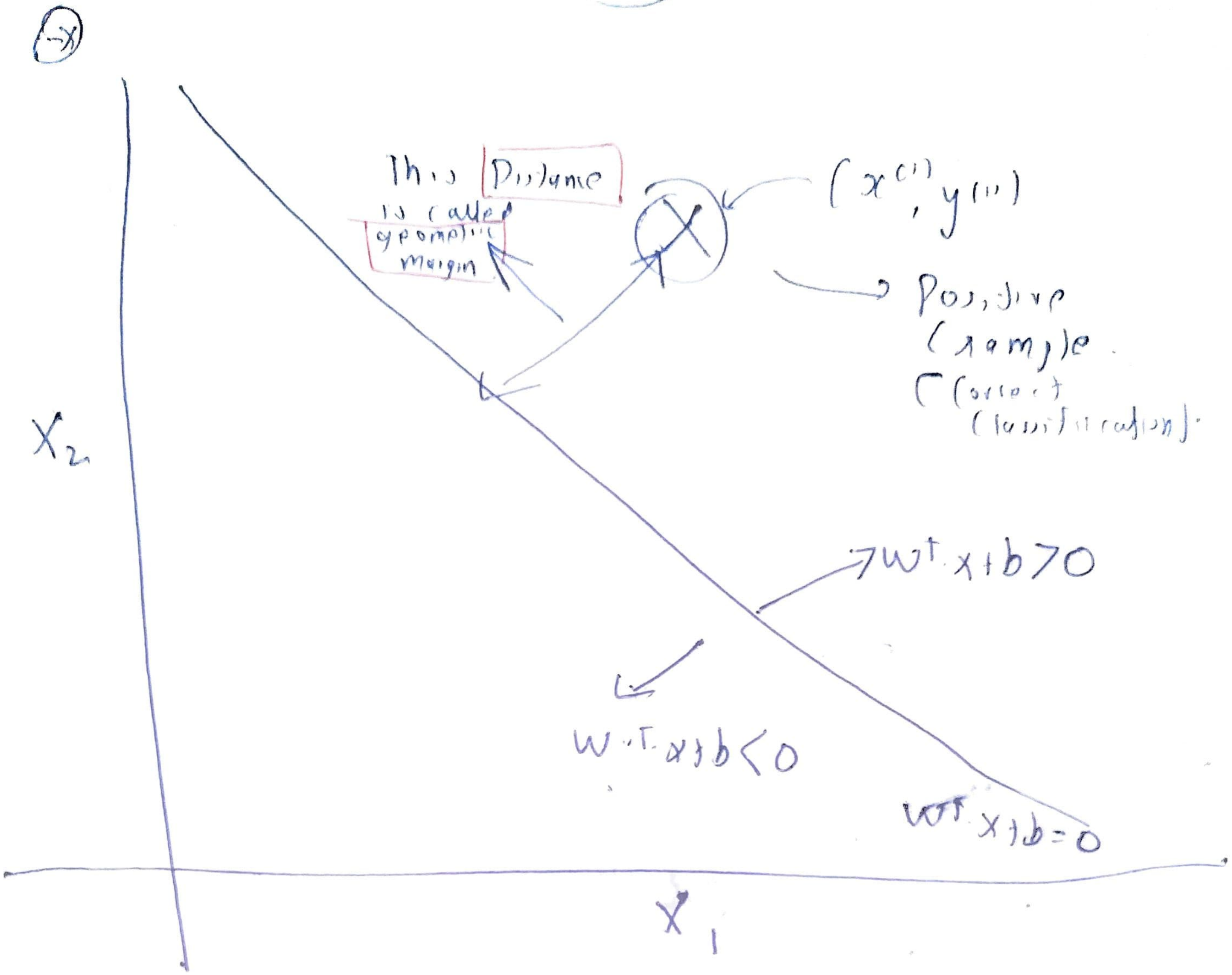
↳ You could normalize the length of your parameters.

↳ Replace (w, b) → $\left(\frac{w}{\|w\|}, \frac{b}{\|b\|}\right)$ (Normalization)

⇓

Boundry remain but cheating involving making the functional margin very high is resolved.

Geometric margin



→ Formally Geometric margin of hyperplane (w, b) w.r.t $(x^{(i)}, y^{(i)})$.

$$\boxed{\gamma^{(i)} = \frac{y^{(i)}(W^T x^{(i)} + b)}{\|W\|}}$$

there is a proof to this!!

→ Relationship between functional & geometric margin

$$\gamma^{(i)} = \frac{\hat{\gamma}^{(i)}}{\|W\|}$$

Geometric margin

Functional margin

* Geometric wrt training set

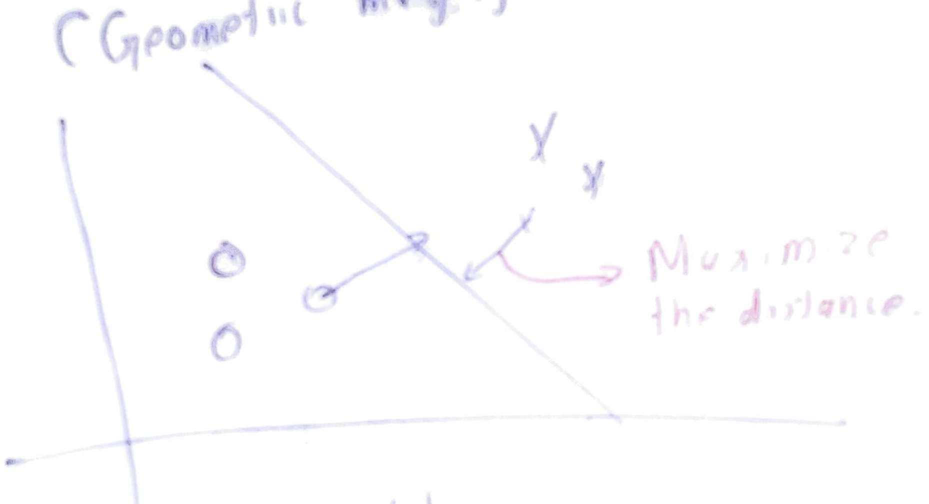
$$\gamma = \min_i \gamma^{(i)}$$

$\hat{\gamma}$ = functional margin

γ = geometric margin

* Optimal margin classifier

- Choose w, b to maximize γ (Geometric margin).



How? (Mathematically)

$$\max_{\gamma, w, b} \gamma$$

$$\text{s.t. } \frac{y^{(i)}(w^T x^{(i)} + b)}{\|w\|} \geq \gamma \quad i = 1, \dots, m$$

⊗ this form is not a convex optimization problem therefore not solvable.

⊗ However it can be reformed:

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

→ Same as before but rewritten and problem is convex optimization in this case.