CS229 Note Lecture 06

Naive Bayes

Recap feature vector

$$X = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} a \\ aardvark \\ ausworth \\ \vdots \\ buy \\ \vdots \\ cs229 \\ \vdots \\ zymurgy \end{matrix}$$

and generative learning algorithm

$$P(X \mid y) = \prod_{i=1}^{n} P(X_i \mid y) \quad \text{and} \quad P(y)$$

using **Bayes Rule** combine these together

$$\arg\max_y P(y \mid X) = \arg\max_y P(X \mid y)P(y)$$

Draw attention on two things

$$X_i \in \{0, 1\} \qquad length(X) = \#\text{words in dictionary}$$

Two Variations on Naive Bayes

1. X_i takes on more values

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

gives us

$$P(X \mid y) = \prod_{i=1}^{n} P(X_i \mid y)$$

so the $P(X_i \mid y)$ must be multinomial probabilities rather than Bernoulli's. This is very common when you need to discretize a continuous value. Predicting the price of houses is a prefect example

Living area	< 500	$1000 - 1500$	$1500 - 2000$	> 2000
$X_i =$	1	2	3	4

2. specific to classifying text documents(generally, for classifying squences)

   The previous model X called **Multivariate Bernoulli Event Model**.

   For **Multinomial Event Model**, represent the i'th feature vector $X^{(i)}$ as

   $$(X_1^{(i)}, X_2^{(i)}, \dots, X_{n_i}^{(i)}) \quad \text{where } n_i = \#\text{words in email}$$

   and each element of this feature vector just written as X_j, X_j will be an index into dictionary

   $$X_j \in \{1, 2, \dots, 50000\} \quad \text{if dictionary has 50000 words}$$

   In this situation, the generative model will be

   $$P(X, y) = \left(\prod_{i=1}^{n} P(X_i \mid y) \right) P(y)$$

   where n is the length of the email.

   So the parameters of this model are

   $$\begin{aligned} \phi_{k|y=1} &= P(X_j = k \mid y) \\ \phi_{k|y=0} &= P(X_j = k \mid 0) \\ \phi_y &= P(y = 1) \end{aligned}$$

   Given a training set, the maximum likelihood estimate of the parameters will be

   $$\phi_{k|y=1} = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = 1\} \sum_{j=1}^{n_i} 1\{X_j^{(i)} = k\}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} \cdot n_i}$$

   Using Laplace Smoothing

$$\phi_{k|y=1} = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} \sum_{j=1}^{n_i} 1\{X_j^{(i)} = k\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} \cdot n_i + (\#\text{words in dictionary})}$$

It turns out Multinomial Event Model always works better than Multivariate Bernoulli Event Model.

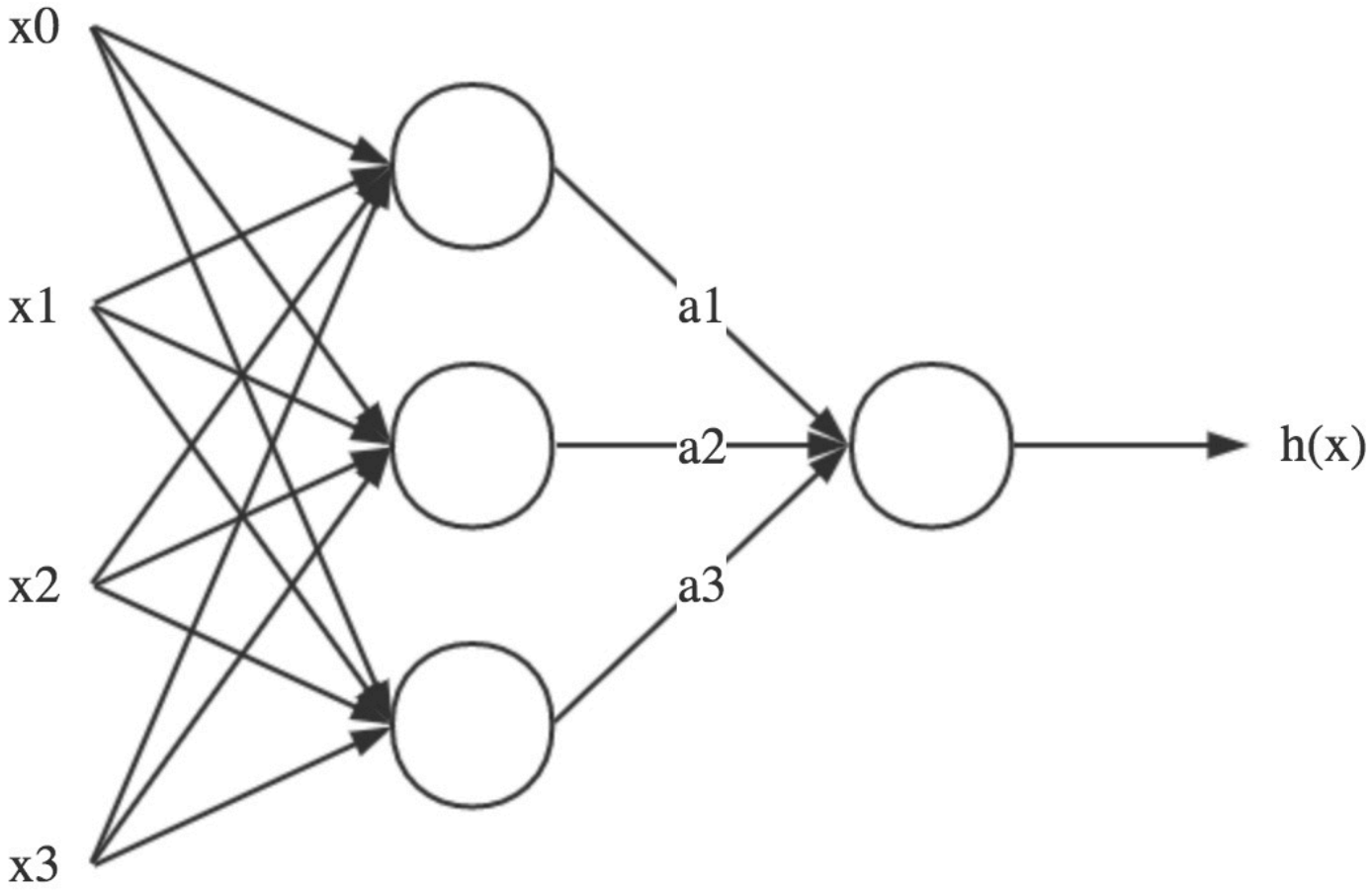
In **Natural Language Processing**, multinomial event model called **Unigram Model**, and there are some other models such as **Higher Order Markov Model**, **Bigram Model** and **Trigram Model**.

And Naive Bayes Model also belongs to Exponential Family。

Maximum Likelihood Estimate

$$\begin{aligned}\ell(\phi_{k|y=1}, \phi_{k|y=0}, \phi_y) &= \log \prod_{i=1}^{m} P(X^{(i)}, y^{(i)}; \phi_{k|y=1}, \phi_{k|y=0}, \phi_y) \\ &= \log \prod_{i=1}^{m} \prod_{j=1}^{n_i} P(X_j^{(i)}, y^{(i)}; \phi_{k|y=1}, \phi_{k|y=0}, \phi_y) \cdot P(y^{(i)}; \phi_y)\end{aligned}$$

Neural Network



$$\vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} g(X^T\theta^{(1)}) \\ g(X^T\theta^{(2)}) \\ g(X^T\theta^{(3)}) \end{bmatrix} \quad h_\theta(x) = g(\vec{a}^{\,T}\theta^{(4)})$$

And the loss function is

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - h_\theta(x^{(i)})\right)$$

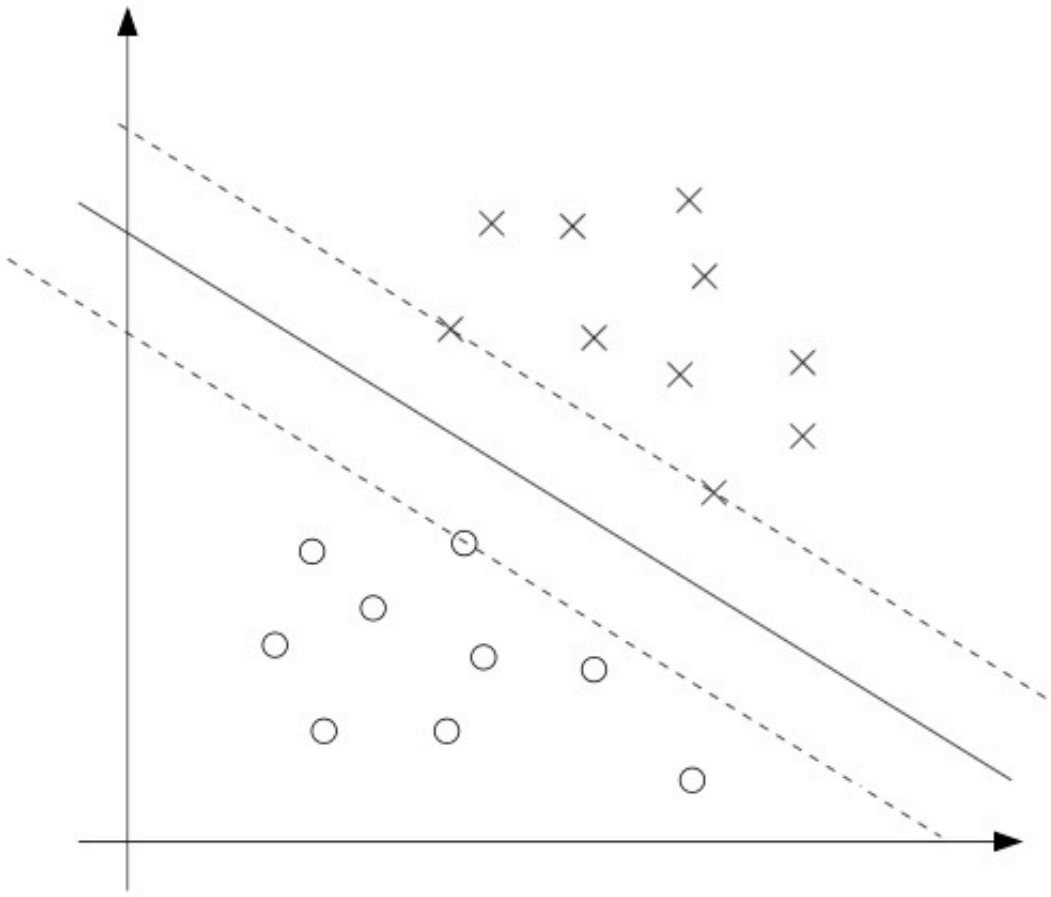
Support Vector Machine(SVM)

We can think of logistic regression as a algorithm:

$$\begin{aligned} &\text{Compute } \theta^T x \\ &\quad \text{Predict "1"} \Leftrightarrow \theta^T x \geq 0 \\ &\quad \text{Predict "0"} \Leftrightarrow \theta^T x < 0 \end{aligned}$$

If $\theta^T x \gg 0$, very "confident" that $y = 1$, and if $\theta^T x \ll 0$, very "confident" that $y = 0$.

Logistic regression model wouldn't be nice if $\forall i$ st. $y = 1$ we have $\theta^T x \gg 0$, and $\forall i$ st. $y = 1$ we have $\theta^T x \ll 0$.



The middle line is a much better linear separator than the others. One reason is the middle is just further from the data. Another reason is the distance between the middle and training examples(**Geometric Margin**)

Notation

$$
\begin{aligned}
&y \in \{-1, +1\} \\
&h \in \{-1, +1\} \\
&g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{Otherwise} \end{cases} \\
&h_\theta(x) = g(\theta^T x) \quad x \in \mathbb{R}^{n+1} \\
&h_{w,b}(x) = g(w^T x + b) \quad x, w \in \mathbb{R}^n, b \in \mathbb{R}
\end{aligned}
$$

Definition

Define **Functional Margin** of the hyperplane (w, b) with respect to a specific training example $(x^{(i)}, y^{(i)})$ is

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

If $y^{(i)} = 1$, want $w^T x + b \gg 0$.
If $y^{(i)} = -1$, want $w^T x + b \ll 0$.
If $y^{(i)}(w^T x^{(i)} + b) > 0$, that means we classified $(x^{(i)}, y^{(i)})$ correctly.

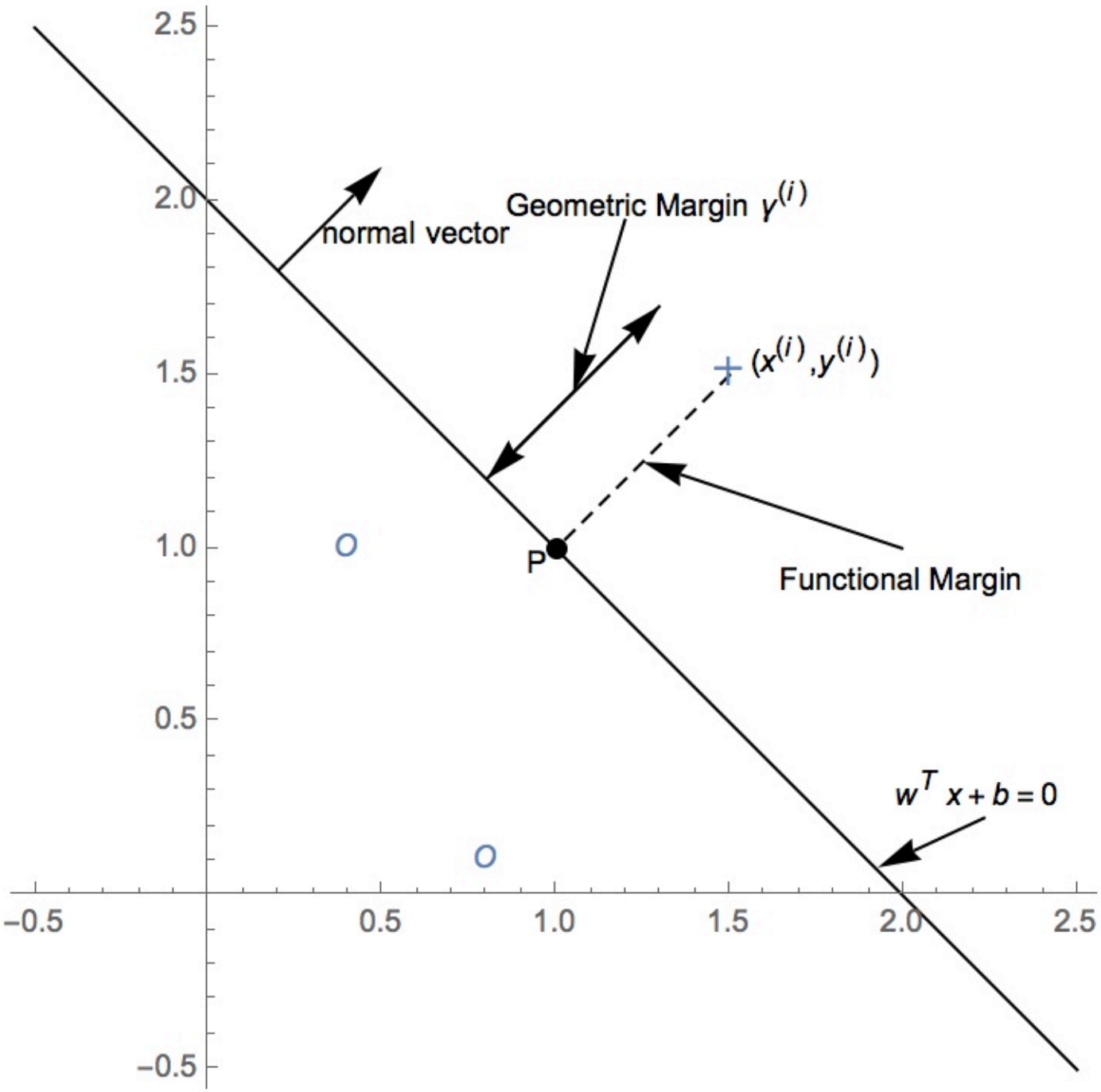
The functional margin of a hyperplane with respect to an entire training set is going to define

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

If we double w and b, then we can easily double functional margin. So this goal of making functional margin large isn't so useful. By adding a normalization condition can solve this problem such as

$$||w|| = 1$$

Geometric Margin



The **Normal Vector** of the hyperplane is

$$\frac{w}{||w||}$$

The Geometric Margin $\gamma^{(i)}$ means that point p is going to be

$$x^{(i)} - \gamma^{(i)} \cdot \frac{w}{||w||}$$

Because point p is on hyperplane, so point p must satisfy

$$w^T(x^{(i)} - y^{(i)}\frac{w}{||w||}) + b = 0$$

Solve this equation for γ

$$\begin{aligned} w^T x^{(i)} + b &= \gamma^{(i)} \cdot \frac{w^T w}{||w||} \\ &= \gamma ||w|| \end{aligned}$$

therefore

$$\gamma^{(i)} = \left(\frac{w}{||w||} \right)^T x^{(i)} + \frac{b}{||w||}$$

So, more generally, to find the geometric margin of a training example to be

$$\gamma^{(i)} = y^{(i)} \left[\frac{w^T}{||w||} x^{(i)} + \frac{b}{||w||} \right]$$

And so a easy fact is if $||w|| = 1$, then $\hat{\gamma}^{(i)} = \gamma^{(i)}$. More generally $\gamma^{(i)} = \frac{\hat{\gamma}^{(i)}}{||w||}$.

Final definition is **Geometric Margin**

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

and so the maximum margin classifier which is a precursor to the SVM. The max margin classifier written as

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