Learning From Data
Lecture 4: Generative Learning Algorithms

Xiangxiang Xu
xuxx14@mails.tsinghua.edu.cn

10/11/2019

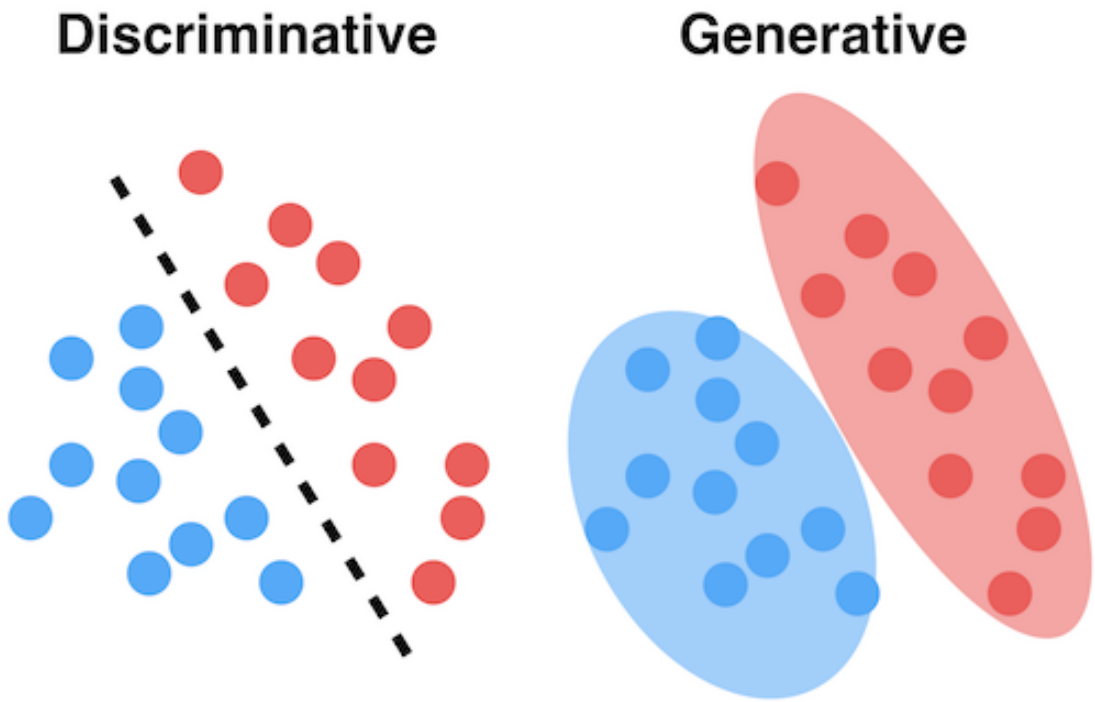
Today's Lecture

Supervised Learning (Part II)

- Discriminative & Generative Models
- Gaussian Discriminant Analysis
- Naïve Bayes

Two Learning Approaches

Classify input data x into two classes $y \in \{0, 1\}$

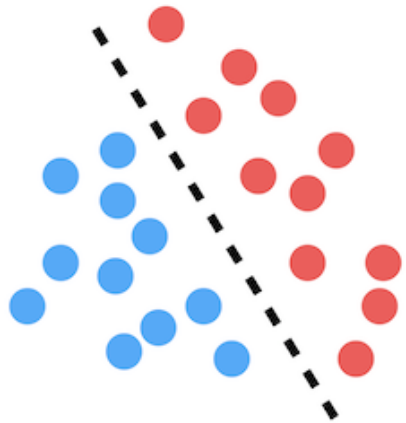


Discriminate between classes of data points

Model the underlying distribution of the data

Discriminative Learning Algorithms

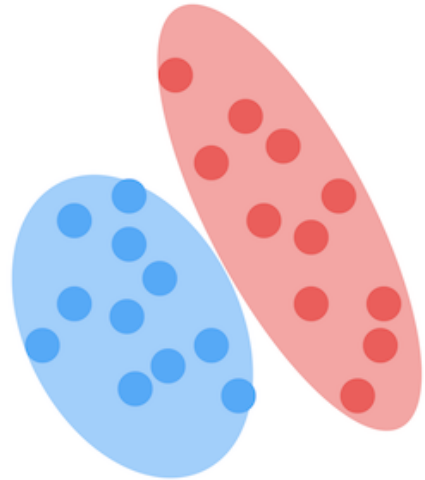
A class of learning algorithms that try to learn the **conditional probability** $p(y|x)$ directly or learn mappings directly from $\mathcal{X}$ to $\mathcal{Y}$.



- e.g. linear regression, logistic regression, k-Nearest Neighbors ...

Generative Learning Algorithms

A class of learning algorithms that model the **joint probability** $p(x, y)$.



- Equivalently, generative algorithms model $p(x|y)$ and $p(y)$
- $p(y)$ is called the **class prior**
- Learned models are transformed to $p(y|x)$ later to classify data using Bayes' rule

Bayes Rule

The posterior distribution on y given x:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

Bayes Rule

The posterior distribution on y given x:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

Make predictions in a generative model:

$$\begin{aligned} \operatorname*{argmax}_y p(y|x) &= \operatorname*{argmax}_y \frac{p(x|y)p(y)}{p(x)} \\ &= \operatorname*{argmax}_y p(x|y)p(y) \end{aligned}$$

No need to calculate $p(x)$.

Generative Models

Generative classification algorithms:

- Continuous input: Gaussian Discriminant Analysis
- Discrete input: Naïve Bayes

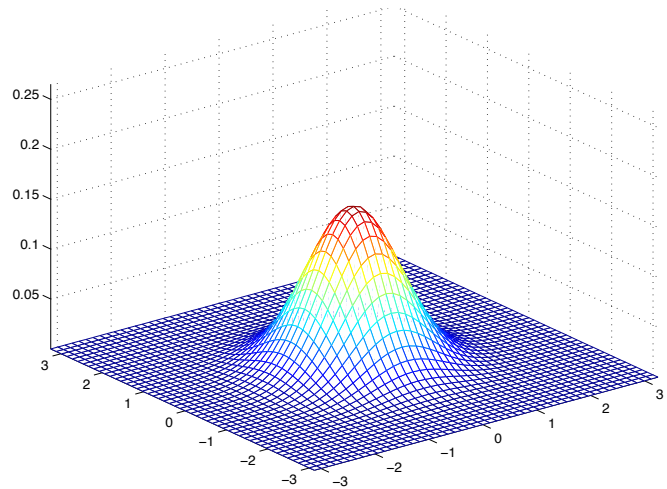
Multivariate Normal Distribution

Multivariate normal (or multivariate Gaussian) distribution $N(\mu, \Sigma)$

- $\mu \in \mathbb{R}^n$ is the mean vector
- $\Sigma \in \mathbb{R}^{n \times n}$ is the covariance matrix. Σ is symmetric and PSD

Density function:

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$



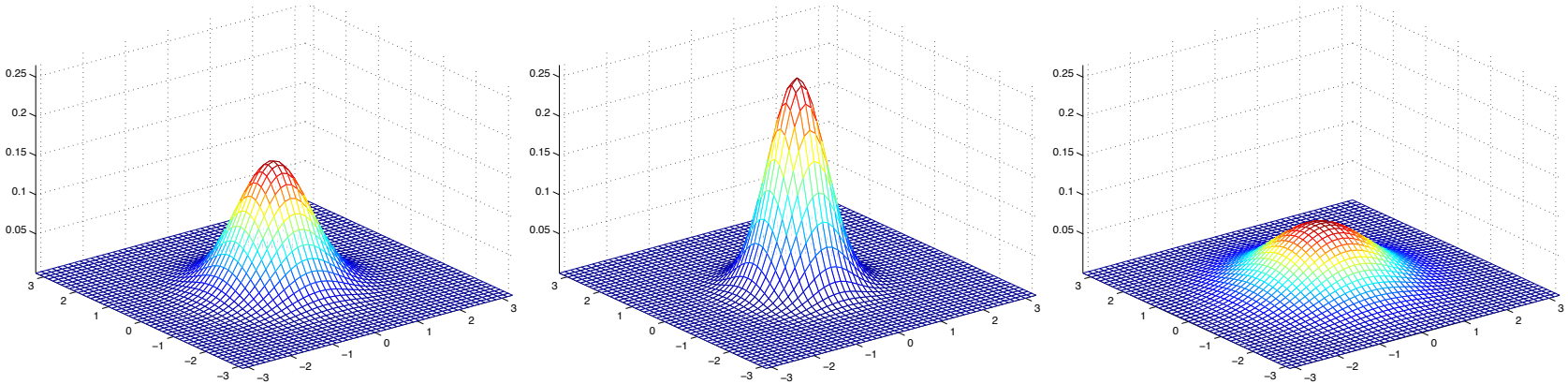
Multivariate Normal Distribution

Let $X \in \mathbb{R}^n$ be a random vector. If $X \sim N(\mu, \Sigma)$,

$$\mathbb{E}[X] = \int p(x; \mu, \Sigma) dx = \mu$$

$$\text{Cov}(X) = \mathbb{E}\left[(X - \mathbb{E}[X])(X - \mathbb{E}[X])^T\right] = \Sigma$$

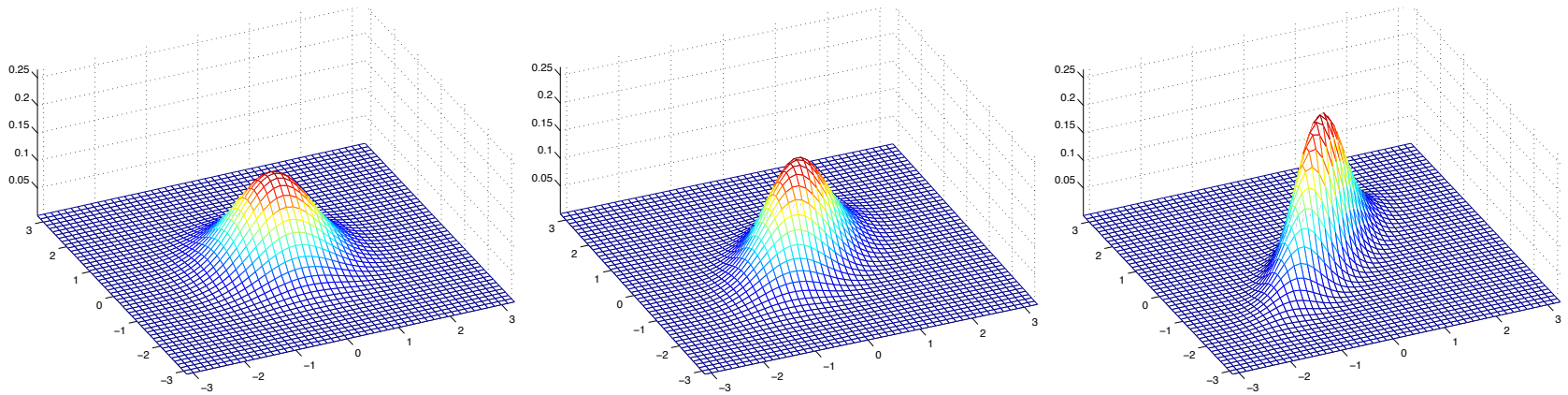
Gaussian Discriminative Analysis



$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Diagonal entries of Σ controls the "spread" of the distribution

Gaussian Discriminative Analysis



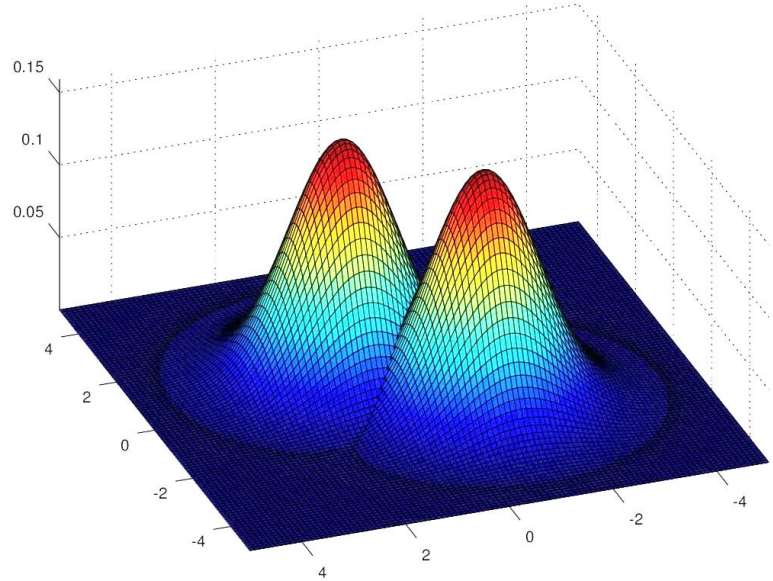
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

The distribution is no longer oriented along the axes when off-diagonal entries of Σ are non-zero.

Gaussian Discriminant Analysis (GDA) Model

Given parameters $\phi, \mu_0, \mu_1, \Sigma$,

$$y \sim \text{Bernoulli}(\phi)$$
$$x|y=0 \sim \mathcal{N}(\mu_0, \Sigma)$$
$$x|y=1 \sim \mathcal{N}(\mu_1, \Sigma)$$



Probability density functions:

$$p(y) = \phi^y (1-\phi)^{1-y}$$

$$p(x|y=0) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\right)$$

$$p(x|y=1) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)$$

Log likelihood of the data:

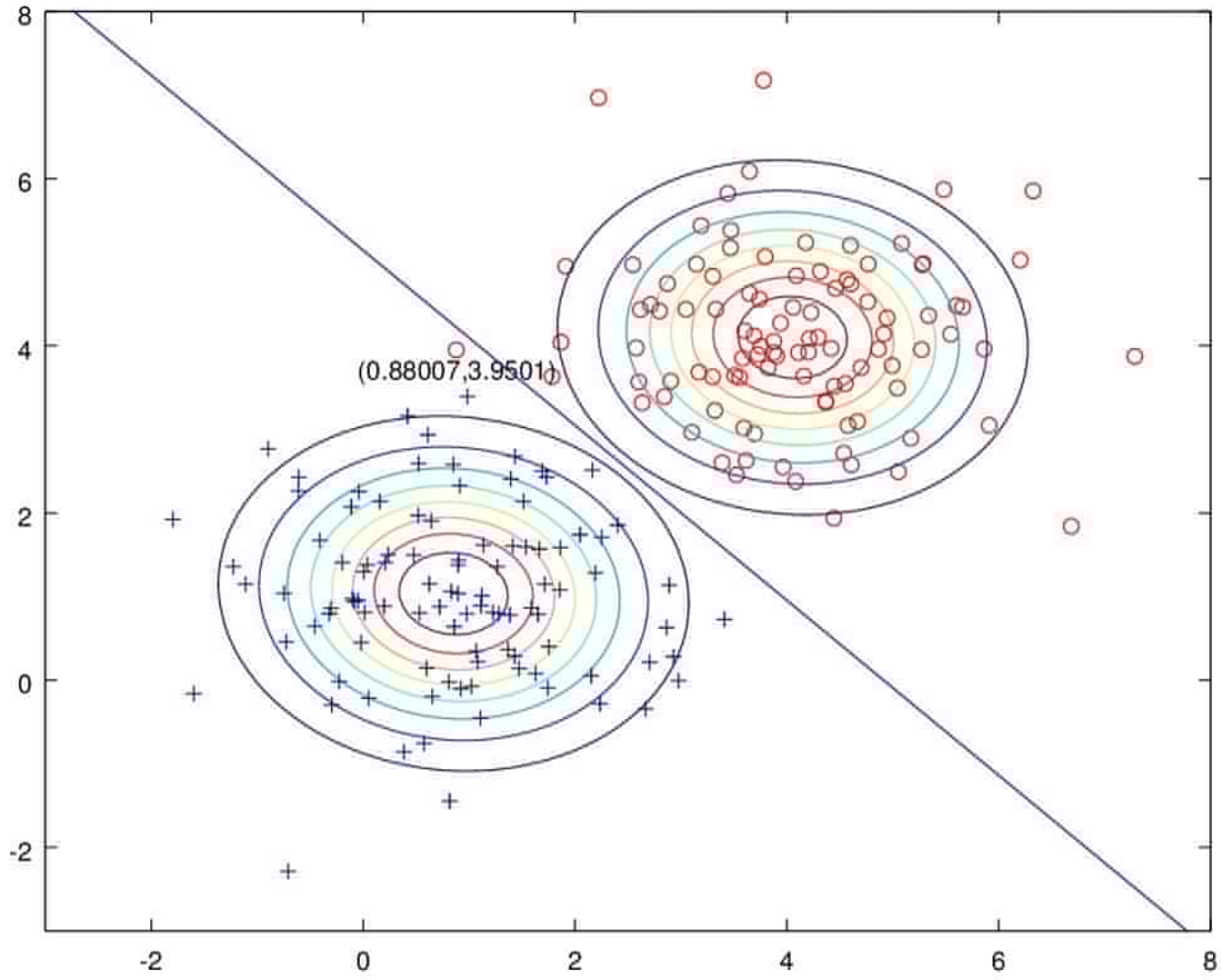
$$
\begin{aligned}
l(\phi, \mu_0, \mu_1, \Sigma) &= \log \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}; \phi, \mu_0, \mu_1, \Sigma) \\
&= \log \prod_{i=1}^{m} p(x^{(i)} | y^{(i)}; \mu_0, \mu_1, \Sigma) p(y^{(i)}; \phi)
\end{aligned}
$$

Maximum likelihood estimate of the parameters:

$$
\begin{aligned}
\phi &= \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\{y^{(i)} = 1\} \\
\mu_b &= \frac{\sum_{i=1}^{m} \mathbf{1}\{y^{(i)} = b\} x^{(i)}}{\sum_{i=1}^{m} \mathbf{1}\{y^{(i)} = b\}} \text{ for } b = 0, 1 \\
\Sigma &= \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T
\end{aligned}
$$

Maximum likelihood estimation of GDA

GDA finds a linear decision boundary at which
$p(y = 1|x) = p(y = 0|x) = 0.5$



GDA and Logistic Regression

$p(y = 1|x; \phi, \mu_0, \mu_1, \Sigma)$ can be written in the form:

$$p(y = 1|x; \phi, \Sigma, \mu_0, \mu_1) = \frac{1}{1 + e^{-\theta^T x}}$$

where $\theta = \begin{bmatrix} \log \frac{1-\phi}{\phi} - \frac{1}{2}(\mu_0^T \Sigma^{-1} \mu_0 - \mu_1^T \Sigma^{-1} \mu_1) \\ \Sigma^{-1}(\mu_0 - \mu_1) \end{bmatrix}$

Hint: the canonical link function of logistic regression $y|x \sim \text{Bernoulli}(\psi)$ is $\log \frac{\psi}{1-\psi}$,

$$\theta^T x = \log \frac{\psi}{1-\psi} = \log \frac{p(y = 1|x)}{p(y = 0|x)} = \log \frac{p(x|y = 1)p(y = 1)}{p(x|y = 0)p(y = 0)}$$

If $p(x|y) \sim \mathcal{N}(\mu, \Sigma)$, $p(y|x)$ is a logistic function.

GDA and Logistic Regression

Logistic Regression

- Maximizes the **conditional likelihood** $\prod_{i=1}^{m} p(y^{(i)}|x^{(i)})$
- Modeling assumptions: $p(y|x)$ is a logistic function; no restriction on $p(x)$
- More robust and less sensitive to incorrect modeling assumptions.

GDA

- Maximizes the **joint likelihood** $\prod_{i=1}^{m} p(x^{(i)}, y^{(i)})$
- Modeling assumptions: $x|y{=}b \sim \mathcal{N}(\mu_b, \Sigma)$, $y \sim \text{Bernoulli}(\phi)$
- When modeling assumptions are correct, GDA is more data efficient (i.e., requires less training data to learn "well")

Naïve Bayes

A simple generative learning algorithm for discrete input variables

Example: Spam filter

Classify email messages x to spam ($y = 1$) and non-spam ($y = 0$) classes.

Binary text features

Given a dictionary of size n, represent a message composed of dictionary words as $x \in \{0, 1\}^n$:

$$x_i = \begin{cases} 1 & i\text{-th dictionary word is in message} \\ 0 & \text{otherwise} \end{cases}$$

$$x = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \\ \vdots \end{bmatrix} \begin{matrix} a \\ aardvark \\ \vdots \\ buy \\ \vdots \end{matrix}$$

Naïve Bayes Model

Probability of observing email $x_1, \ldots, x_n$ given spam class y :

$$p(x_1, \ldots, x_n|y) = p(x_1|y)p(x_2|y, x_1), \ldots, p(x_n|y, x_1, \ldots, x_{n-1})$$

Naïve Bayes (NB) assumption

x_i's are conditionally independent given y:

$$p(x_i|y, x_1, \ldots, x_{i-1}) = p(x_i|y)$$

$$p(x_1, \ldots, x_n|y) = p(x_1|y)p(x_2|y) \ldots p(x_n|y) = \prod_{i=1}^{n} p(x_i|y)$$

Naïve Bayes Parameters

Multi-variate Bernoulli event model

$$p(x, y) = p(y)p(x|y) = p(y)\prod_{i=1}^{n} p(x_i|y)$$

- Assume email class (spam vs no-spam) is randomly generated with prior $p(y = 1) = \phi_y$
- Given y, each word x_i is included in the message independently with $p(x_i = 1|y) = \phi_{i|y}$

Model parameters:

- $\phi_y = p(y = 1)$
- $\phi_{i|y=1} = p(x_i = 1|y = 1)$
- $\phi_{i|y=0} = p(x_i = 1|y = 0)$

Naïve Bayes Parameter Learning

Likelihood of training data $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$:

$$L(\phi_y, \phi_{j|y=0}, \phi_{j|y=1}) = \prod_{i=1}^{m} p(x^{(i)}, y^{(i)})$$

Maximum likelihood estimation of parameters:

$$\phi_y = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\{y^{(i)} = 1\}$$

$$\phi_{j|y=b} = \frac{\sum_{i=1}^{m} \mathbf{1}\{x_j^{(i)} = 1, y^{(i)} = b\}}{\sum_{i=1}^{m} \mathbf{1}\{y^{(i)} = b\}} \text{ for } b = 1, 0$$

Naïve Bayes Prediction

Given new example with feature x, compute the posterior probability

$$
\begin{aligned}
p(y=1|x) &= \frac{p(x|y=1)p(y=1)}{p(x)} \\
&= \frac{p(x|y=1)p(y=1)}{p(x|y=1)p(y=1)+p(x|y=0)p(y=0)} \\
&= \frac{\prod_{i=1}^{n} p(x_i|y=1)p(y=1)}{\prod_{i=1}^{n} p(x_i|y=1)p(y=1)+\prod_{i=1}^{n} p(x_i|y=0)p(y=0)}
\end{aligned}
$$

Choose label $y=1$ if $p(y=1|x) > 0.5$

Laplace smoothing

Issue with Naïve Bayes prediction:

- Suppose word x_j hasn't been seen in the training data, $\phi_{j|y=1} = \phi_{j|y=0} = 0$
- Can not compute class posterior $p(y = 1|x) = \frac{0}{0}$.

Let $z \in \{1, \ldots, k\}$ be a multinomial random variable. Given m independent observations $z^{(1)} \ldots z^{(m)}$, maximum likelihood estimation of $\phi_j = p(z = j)$ with **Laplace smoothing** is

$$\phi_j = \frac{\sum_{i=1}^{m} 1\{z^{(i)} = j\} + 1}{m + k}$$

- $\phi_j \neq 0$ for all j
- $\sum_{j=1}^{k} \phi_j = 1$

Naïve Bayes with Laplace smoothing

Apply Laplace smoothing to $\phi_{j|y=b}$ for $b \in \{0, 1\}$

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

In practice we don't apply Laplace smoothing to $\phi_y = p(y = 1)$, which is greater than 0.

Naïve Bayes and Multinomial Event Model

Alternative text representation

- $x_i \in \{1, \ldots, |V|\}$ where $|V|$ is the dictionary size
- Represent email of n words as $x = \{x_1, \ldots, x_n\}$

"a free gift ..." $\rightarrow$ $\{x_1 = 1, x_2 = 1300, x_3 = 2433, \ldots\}$

dictionary id	1	2	...	1300	...	2433	...
word	a	aa	...	free	...	gift	...

Multinomial event model

- assume spam/non-spam is determined randomly by $p(y)$
- Select $x_1, x_2, \ldots, x_n$ independently from the same Multinomial distribution
- Overall probability: $p(x_1, \ldots, x_n, y) = p(y) \prod_{i=1}^{n} p(x_i|y)$

Multinomial event model parameters

Assume $p(x_j = k|y)$ is the same for all j

- $\phi_y = p(y = 1)$
- $\phi_{k|y=1} = p(x_j = k|y = 1)$ for any $j \in \{1, \ldots, n\}$
- $\phi_{k|y=0} = p(x_j = k|y = 0)$ for any $j \in \{1, \ldots, n\}$

Likelihood of training set $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$:

$$\begin{aligned} L(\phi_y, \phi_{k|y=0}, \phi_{k|y=1}) &= \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}) \\ &= \prod_{i=1}^{m} p(y^{(i)}; \phi_y) \prod_{j=1}^{n_i} p(x_j{}^{(i)}|y; \phi_{k|y=0}, \phi_{k|y=1}) \end{aligned}$$

where n_i is the # words in the i-th email.

Maximum likelihood estimation with Laplace smoothing

- $\phi_y = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\{y^{(i)} = 1\}$
- $\phi_{k|y=1} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i} \mathbf{1}\{x_j^{(i)} = k, y^{(i)} = 1\} + 1}{\sum_{i=1}^{m} \mathbf{1}\{y^{(i)} = 1\} n_i + |V|}$
- $\phi_{k|y=0} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i} \mathbf{1}\{x_j^{(i)} = k, y^{(i)} = 0\} + 1}{\sum_{i=1}^{m} \mathbf{1}\{y^{(i)} = 0\} n_i + |V|}$