Learning From Data
Lecture 4: Generative Learning Algorithms

Yang Li yangli@sz.tsinghua.edu.cn

March 22, 2024

Today's Lecture

Supervised Learning (Part II)

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

Ask me a question

Stochastic gradient descent?

cost of each sample

$$\min_\theta J(\theta) = \sum_{i=1}^{m} J_i(\theta) \longrightarrow$$

sampling without replacement

Q1

In SGD, is the randomness from selecting a single data point randomly within an epoch, or from shuffling the data and then starting from a random point to traverse?

$\nabla_{\theta_j} J \longrightarrow \nabla_{\theta_j} J_i(\theta) \rightarrow$ single training sample.

estimate it with

Batch GD.

$$\theta := \theta - \alpha \nabla J(\theta)$$

$$\hookrightarrow \begin{bmatrix} \nabla_{\theta_1} J(\theta) \\ \vdots \\ \nabla_{\theta_n} J(\theta) \end{bmatrix}$$

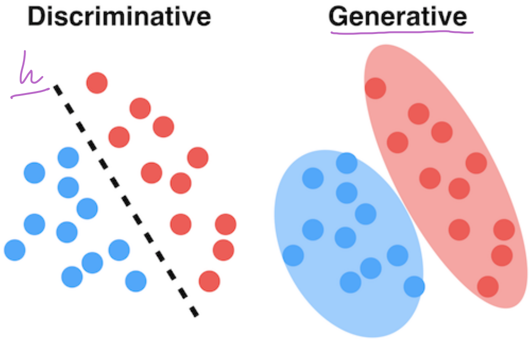
Discriminative & Generative Models

Two Learning Approaches

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

generative model learning

decision function → h



判别

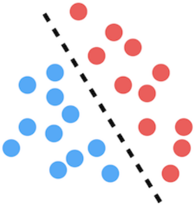
Discriminate between classes of data points	Model the underlying distribution of the data

↳ derive $P(y|x)$

↳ sample from $P(x, y)$

Discriminative Learning Algorithms

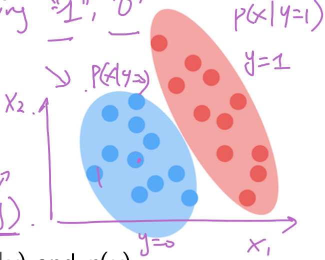
A class of learning algorithms that try to learn the **conditional probability** $p(y|x)$ directly or learn mappings directly from $\mathcal{X}$ to $\mathcal{Y}$. $h(x)$



$h: X \to Y$.

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

classifying "1", "0"



Generative Learning Algorithms

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

input label. $p(x,y) = p(x|y)\,p(y)$.

- 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:

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

unnormalize probability.

No need to calculate $p(x)$.

Generative Models

Generative classification algorithms:

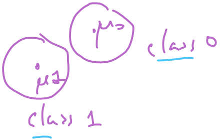
- Continuous input: Gaussian Discriminant Analysis
- Discrete input: Naïve Bayes

Gaussian Discriminant Analysis

↳ linear discriminant analysis (LDA)

↳ quadratic discriminant analysis (QDA).

Gaussian Discriminant Analysis: Overview



Goal

Binary classification with input in $\mathcal{X} = \mathbb{R}^n$ and label in $\mathcal{Y} = \{0, 1\}$

$p(y)$ – binary
$p(x|y)$ – continuous

Main steps

1. Select a *data generating distribution* .

$x \in \mathbb{R}^n$

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

$p(y)$

when Σ is not shared.
(Σ_1, Σ_2)

2. Estimate model parameters ϕ, μ_0 ,μ_1 and Σ from training data.
3. For any new sample x', predict its label by computing $p(y|x = x'; \phi, \mu_0, \mu_1, \Sigma)$

Multivariate Normal Distribution

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

- $\underline{\mu} \in \mathbb{R}^n$ is the mean vector,
- $\underline{\Sigma} \in \mathbb{R}^{n \times n}$ is the covariance matrix. <u>Σ is symmetric and SPD.</u>

semi-positive definite

$\forall x \neq 0, \quad x^T \underline{\Sigma} x \geq 0$

Density function:

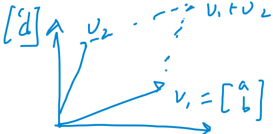
$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} \underbrace{|\Sigma|^{1/2}}_{\det(\Sigma)}} e^{\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)}$$

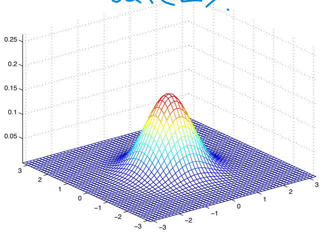
$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{matrix} v_1 \\ v_2 \end{matrix}$$

$\det(A) = \underline{ad - cd}$.

$\mu \in \mathbb{R}^2$

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}.$$





$\det(A) = \text{Area}(\square)$.

Multivariate Normal Distribution

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

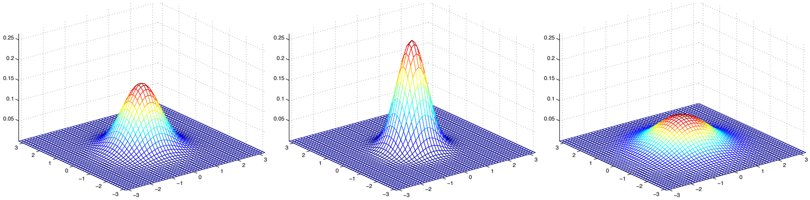
$$\mathbb{E}[X] = \int_x x \, p(x; \mu, \Sigma) dx = \mu$$

$$\mathrm{Cov}(X) = \mathbb{E}\left[(X - \underbrace{\mathbb{E}[X]}_{\mu})(X - \underbrace{\mathbb{E}[X]}_{\mu})^T\right] = \Sigma$$

Gaussian Discriminative Analysis

$\mathcal{N}(\mu, \sigma I)$

Isotropic Gaussian Distribution



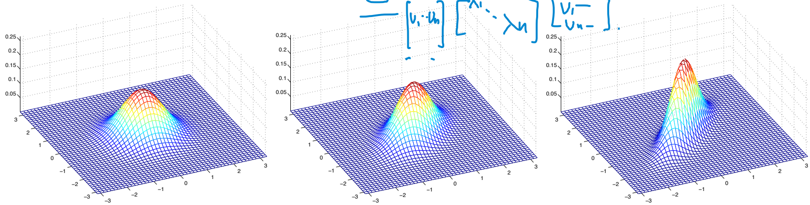
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

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

Gaussian Discriminative Analysis

$v_2 \quad v_1$

$$\Sigma = U \Lambda U^T$$
$$= \begin{bmatrix} u_1 \cdots u_n \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \begin{bmatrix} u_1 - \\ u_n - \end{bmatrix}$$



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

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

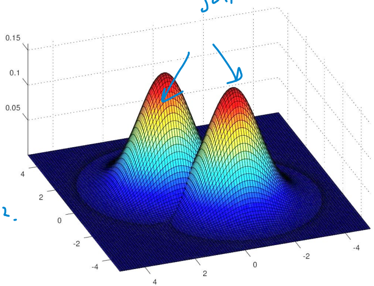
Gaussian Discriminant Analysis (GDA) Model

same covariance

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

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

$x \in \mathbb{R}^2$.



Probability density functions:

$$p(y) = \underline{\phi^y(1-\phi)^{1-y}}$$
$$p(x|y=0) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\right)}$$
$$p(x|y=1) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)}$$

Log likelihood of the data: $p(x, y; \phi, \mu_0, \mu_1, \Sigma)$

joint all sample $\in \mathbb{R}$, $\in \mathbb{R}^n$, $\in \mathbb{R}^{n \times n}$

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

$\phi^{y_i}(1-\phi)^{1-y_i}$

$\mu_{y_i} = \begin{cases} \mu_0 & \text{if } y_i = 0 \\ \mu_1 & \text{if } y_i = 1 \end{cases}$

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

$$= \sum_{i=1}^{m} \log p(x^{(i)} | y^{(i)}; \mu_0, \mu_1, \Sigma) + \sum_{i=1}^{m} \log p(y^{(i)}; \phi).$$

$$l = \sum_{i=1}^{m} \log \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} - \frac{1}{2}(x^{(i)} - \mu_{y_i})^T \Sigma^{-1} (x^{(i)} - \mu_{y_i}) + \sum_{i=1}^{m} y^{(i)} \log \phi + (1 - y^{(i)}) \log(1-\phi)$$

① Find ϕ^*

$\frac{\partial L}{\partial \phi} = 0$.

$\phi = ?$

② Find μ_0^*

$\nabla_{\mu_0} L = 0$

$\nabla_x (x^T A x)$, A is symmetric

$= 2Ax$

③ Find Σ^*.

$\nabla_\Sigma L = 0$.

$\nabla_A |A| = |A| (A^{-1})^T$

$\nabla_A x^T A^{-1} y = -A^{-T} x y^T A^{-T}$

Log likelihood of the data:

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

Log likelihood of the data:

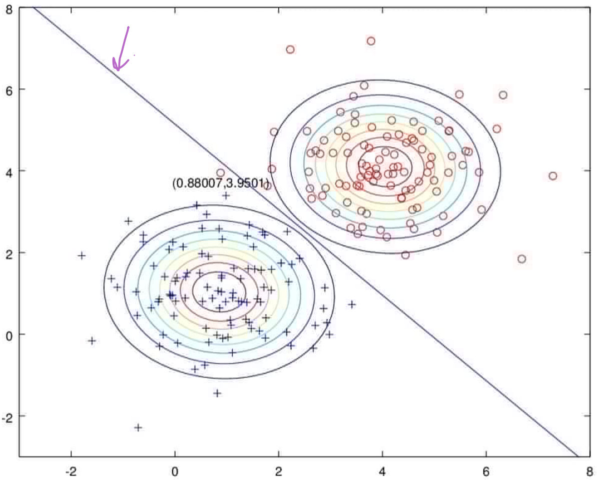
$$l(\phi, \mu_0, \mu_1, \Sigma) = \log \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}; \phi, \mu_0, \mu_1, \Sigma)$$
$$= \log \prod_{i=1}^{m} p(x^{(i)} | y^{(i)}; \mu_0, \mu_1, \Sigma) p(y^{(i)}; \phi)$$

Maximum likelihood estimate of the parameters:

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

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

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

$p(y = 1|x; \phi, \mu_0, \mu_1, \Sigma)$ can be written in the form: logistic $y|x \sim \mathcal{N}$.

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

GDA and Logistic Regression

LDA is a special form of logistic regression

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

$$p(y = 1|x; \underbrace{\phi, \Sigma, \mu_0, \mu_1}_{H}) = \frac{1}{1 + e^{-\theta^T x}} \qquad \theta^T x = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \\ \theta_0 \end{bmatrix}$$

$n+1$ $\quad \theta_1 \quad \theta_0$

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \Sigma^{-1}(\mu_1 - \mu_0) \\ \underbrace{\frac{1}{2}(\mu_0^T \Sigma^{-1} \mu_0 - \mu_1^T \Sigma^{-1} \mu_1) - \log \frac{1-\phi}{\phi}}_{\in \mathbb{R}} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ 1 \end{bmatrix}$$

$n \times 1$

T

~~intercep~~ bias term

$$p(y=1|x;H) = \frac{p(x|y=1;H)\,p(y=1;H)}{p(x|y=1;H)\,p(y=1;H) + p(x|y=0;H)\,p(y=0;H)}$$

$p(x) \rightsquigarrow$

$$= \frac{1}{1 + \exp\left(\underbrace{-(\mu_1 - \mu_0)\Sigma^{-1}}_{\theta_1 \in \mathbb{R}^n} x \underbrace{- \frac{1}{2}(\mu_0^T \Sigma^{-1} \mu_0 - \mu_1^T \Sigma^{-1} \mu_1) - \log \frac{1-\phi}{\phi}}_{\theta_2 \in \mathbb{R}}\right)}$$

GDA and Logistic Regression

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

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

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \Sigma^{-1}(\mu_1 - \mu_0) \\ \frac{1}{2}(\mu_0^T \Sigma^{-1} \mu_0 - \mu_1^T \Sigma^{-1} \mu_1) - \log \frac{1-\phi}{\phi} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ 1 \end{bmatrix}$$

Similarly,

$$p(y = 0|x; \phi, \Sigma, \mu_0, \mu_1) = \frac{1}{1 + e^{\theta^T x}}$$

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

x.

p(y|x) ~ logistic

p(x|y) ~ N(μ, Σ).

LDA (GDA) has strict model assumptions

GDA and Logistic Regression

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 **asymptotically efficient** and **data efficient**

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.

Naïve Bayes

Naïve Bayes: Motivationg Example

A simple generative learning algorithm for discrete input variables

Example: Spam filter (document classification)

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

> **Hello** [redacted]
> We need to confirm your info...
> (1) FINAL MESSAGE: Payout Verification - $3000 PAYOUT is ready to be addressed in your Name and we want to be sure it gets to the right place. Click below to start the confirmation process. The sooner you act, the sooner it can be in your hands!
> Raging Bull Casino

A sample spam email

Example: Spam Filter

document feature
(message)

Binary text features

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

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

$$x = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \mathit{a} \\ \mathit{aardvark} \\ \vdots \\ \mathit{casino} \\ \vdots \\ \mathit{payout} \\ \vdots \\ \mathit{zyzzyva} \end{array} \quad \begin{array}{c} A \\ \\ \\ \text{appears at least once in message} \\ \\ \\ \\ Z \end{array}$$

$\in \{0,1\}^n$

Naïve Bayes Model

$\boxed{P(x|y)}$ $P(y)$ $\}$ $P(x,y)$.

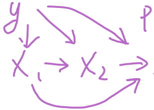
first word, last word

$P(x|y=1) \Rightarrow P_1(x_1,\dots,x_n) = P_1(x_1)P_1(x_2|x_1)P_1(x_3|x_1,x_2)\dots P_1(x_n|x_1,\dots,x_{n-1})$.

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

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

$y = 1$

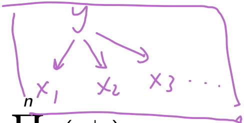


probabilistic graphical model

Naïve Bayes (NB) assumption

x_i's are conditionally independent given y:

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



probability of document with $x_1 \dots x_n$ given y

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

Naïve Bayes Parameters

Multi-variate Bernoulli event model

$x|y$ generated from n independent Bernoulli trials

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

- $y \sim Bernoulli(\phi_y)$: assume email class (spam vs no-spam) is randomly generated with prior $p(y) = \phi_y^y(1-\phi_y)^{1-y}$
- $x_i|y = b \sim Bernoulli(\phi_{i|y=b}), b = 0, 1$: given $y = b$, each word x_i is included in the message independently with $p(x_i = 1|y = b) = \phi_{i|y=b}$. i.e.

$$p(x_i|y = b) = \phi_{i|y=b}^{x_i}(1-\phi_{i|y=b})^{1-x_i}$$

Model parameters:

- ϕ_y $\in \mathbb{R}$
- $\phi_{i|y=1}, \phi_{i|y=0}$ for $i = 1, \ldots, n$

$\}$ $2n+1$

spam non-spam

Naïve Bayes Parameter Learning

Likelihood of i.i.d. 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:

$$\phi_y = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\{y^{(i)} = 1\}$$ *% of spam emails* $\frac{1}{m}\sum_{i=1}^{m} y_i$

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

% of spam(non-spam) emails containing jth dictionary word

Naïve Bayes Prediction

Given new example with feature x, compute the posterior probability

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

$p(x_1, .. x_n|y)$

naive bayes

Choose label $y = 1$ (spam) if $p(y=1|x) > T$ where $T \in [0,1]$ is a threshold .. e.g. $T = 0.5$ — hyperparameter.

T tradeoff between wrongly blocked non-spam (FPs) vs. wrongly blocked spams (FNs).

Laplace smoothing

Issue with Naïve Bayes prediction:

- Suppose word x_j hasn't been seen in the training data,

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{m} 1\{y^i=1, x_j^i=1\}}{\sum_{i=1}^{m} 1\{y^i=1\}} = 0.$$

$$\phi_{j|y=0} = 0$$

Inference: Given x, predict y.

$$p(y=1|x) = \frac{\prod_{l=1}^{n} p(x_l|y=1)p(y=1)}{\prod_{l=1}^{n} \underbrace{p(x_l|y=1)}_{=0 \text{ when } x_l = x_j} p(y=1) + \prod_{l=1}^{n} \underbrace{p(x_l|y=0)}_{=0.} p(y=0)} = \frac{0.}{0.}$$

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 $p(y = 1|x) = \frac{0}{0}$.

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 $p(y=1|x) = \frac{0}{0}$.

Laplace smoothing

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

$\frac{1}{m} \sum_{i=1}^{m} 1\{z^i = j\}$

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

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

check:

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

Naïve Bayes with Laplace smoothing

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

$p(x_j|y)$

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

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

$p(y=0)$

Naïve Bayes Summary

Naïve Bayes (NB) assumption

x_i's are conditionally independent given y:

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

Different event models:

- **Multi-variate Bernoulli model:** represent a document of dictionary size n as n independent Bernoulli trails.
- **Multinomial event model:** represent document of n words as $x = \{x_1, \dots, x_n\}$ where $x_i = \{1, \dots, K\}$ and K is the dictionary size *(optional)*

Naïve Bayes and Multinomial Event Model

Alternative text representation

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

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

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

$$\begin{bmatrix} 1 \\ 1300 \\ 2433 \end{bmatrix}$$

Naive Bayes and Multinomial Event Model

Multinomial

$\phi_{1300|y=b}$. $\leftarrow$ $p(x_1=1300|y=b) = p(x_3=1300|y=b)$

free gift. free money.

$j=1$ $j=3$.

Multinomial event model

- first sampling $y \in \{0,1\}$ from $p(y)$

$$y \sim Bernoulli(\phi_y)$$ 1 parameter

- Select $x_1, x_2, \ldots, x_n$ independently from the same Multinomial distribution $p(x_i|y)$

$$x_i|y = b \sim Multinomial(\phi_{1|y=b}, \ldots, \phi_{K|y=b}), b = 0, 1$$
$$\phi_{k|y=b} = p(x_j = k|y = b) \text{ for all } j \in \{1, \ldots, n\}$$ 2k parameters

For any word k in the dictionary, $\phi_{k|y}$ is the probability of k appear in an email given email class y

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

Multinomial event model parameters

$\rightarrow p(x_j = k|y)$ ← constant.

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 $k = 1, \ldots, K$
- $\phi_{k|y=0} = p(x_j = k|y = 0)$ for $k = 1, \ldots, K$

$2K+1$

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

$$
\begin{aligned}
L(\phi_y, \phi_{k|y=0}, \phi_{k|y=1}) &= \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}) \\
&= \prod_{i=1}^{m} p(x_1^{(i)}, \ldots, x_{n_i}^{(i)}, y^{(i)}) \\
&= \prod_{i=1}^{m} p(y^{(i)}; \phi_y) \prod_{j=1}^{n_i} p(x_j^{(i)} | y; \phi_{k|y=0}, \phi_{k|y=1})
\end{aligned}
$$

words in email. (jth)

m document.

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

Maximum likelihood estimation with Laplace smoothing

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

K is the dictionary size.