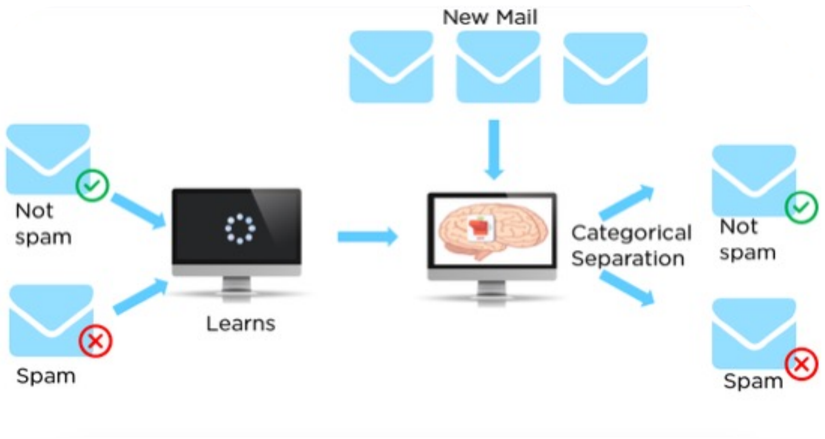
Section 9 Naive Bayes

1. Text classification (Bag of words)
2. Navis Bayes
3. Text classification (Event model)

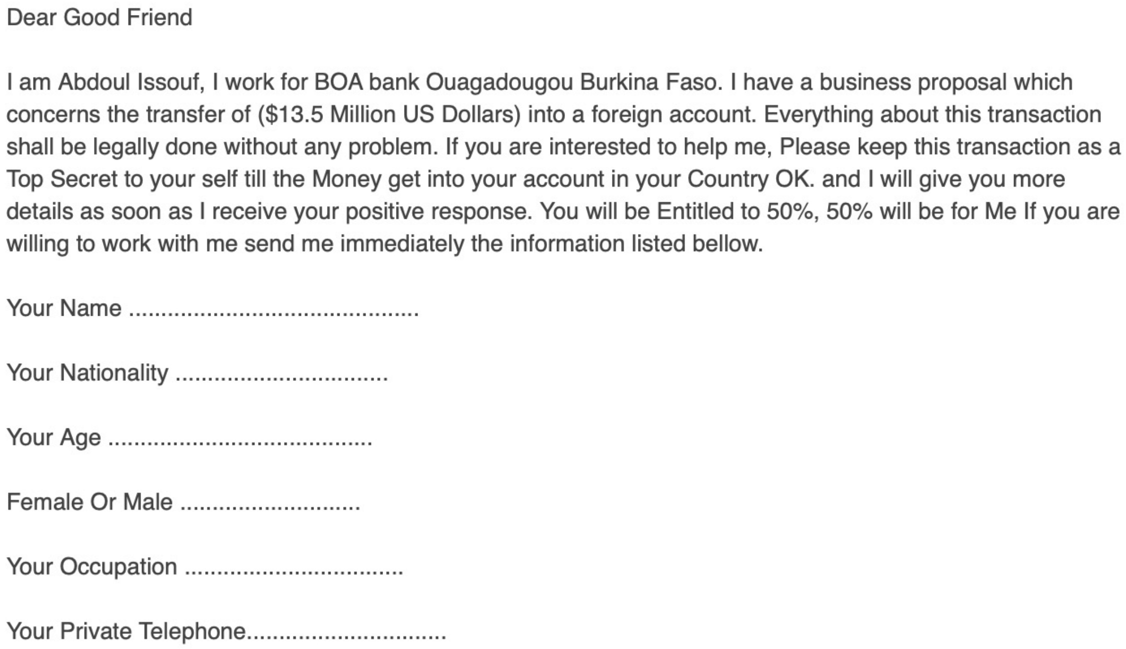
- **Text Classification.**

Example: Separate emails as Spam=0 and Not Spam=1



Example: Twitter, Messages, Facebook, Google, Amazon, …

- Spam Email Sample:



Dear Good Friend

I am Abdoul Issouf, I work for BOA bank Ouagadougou Burkina Faso. I have a business proposal which concerns the transfer of ($13.5 Million US Dollars) into a foreign account. Everything about this transaction shall be legally done without any problem. If you are interested to help me, Please keep this transaction as a Top Secret to your self till the Money get into your account in your Country OK. and I will give you more details as soon as I receive your positive response. You will be Entitled to 50%, 50% will be for Me If you are willing to work with me send me immediately the information listed bellow.

Your Name ....

Your Nationality ....

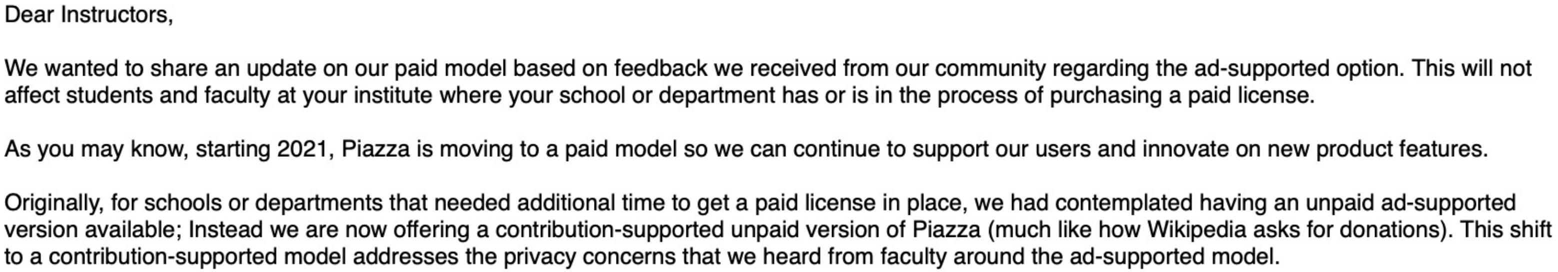
Your Age ....

Female Or Male ....

Your Occupation ....

Your Private Telephone..............................

- Not Spam Email Sample



Dear Instructors,

We wanted to share an update on our paid model based on feedback we received from our community regarding the ad-supported option. This will not affect students and faculty at your institute where your school or department has or is in the process of purchasing a paid license.

As you may know, starting 2021, Piazza is moving to a paid model so we can continue to support our users and innovate on new product features.

Originally, for schools or departments that needed additional time to get a paid license in place, we had contemplated having an unpaid ad-supported version available; Instead we are now offering a contribution-supported unpaid version of Piazza (much like how Wikipedia asks for donations). This shift to a contribution-supported model addresses the privacy concerns that we heard from faculty around the ad-supported model.

Data vectorization: (Bag of words)

We will represent an email via a **feature vector** $\vec{x} \in \mathbb{Z}_2^d$, called vocabulary, whose length *d* is equal to the number of words in the dictionary, e.g., d=171,146.

In practice, we should build a better **dictionary** with only “medium frequency” words, say $d = 2000$. For example,

B={ability, absolute, abuse, access, accident, ..., young, yourself, zip }

The idea is similar to the coordinates in linear algebra:

Let $V = \{\vec{v}\}$. Here, $\vec{v}$ is the set of words of an email. (unordered, unrepeated)

Let dictionary be a **basis B** for V.

Consider the coordinate map relative the dictionary basis:

$$\begin{tikzcd} V \arrow[r] & \mathbb{Z}_2^d \end{tikzcd}$$

For example, we can transfer an email $\vec{v}$ as a vector $\vec{x} \in \mathbb{R}^d$.

$$\vec{v} = \{\text{ability, …, buy, …, help ,…}\} \longrightarrow \vec{x} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Here $x_i = \mathbb{I}(i\text{-th dictionary word in } \vec{v})$

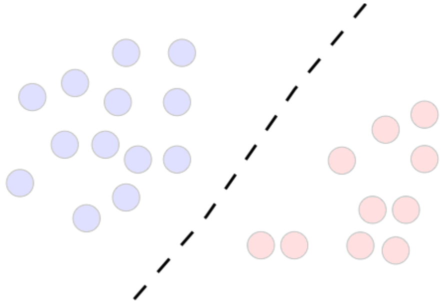
Then the labeled training emails can be represented as our standard

Training data: $\mathcal{D} = \left\{ \left(\vec{x}^{(i)}, y^{(i)} \right) \middle| \, i = 1, \dots, n \right\}$ $\quad y^{(i)} \in \{0,1\} \; or \; \{1, 2, \dots, K\}$

Data matrix: $X = \begin{bmatrix} {\vec{x}^{(1)}}^T \\ {\vec{x}^{(2)}}^T \\ \vdots \\ {\vec{x}^{(n)}}^T \end{bmatrix}$ $\quad \vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$

Methods for classification we learned:

- **Discriminative learning**



$$P(Y = k\,|\vec{X} = \vec{x}) := some\ model$$

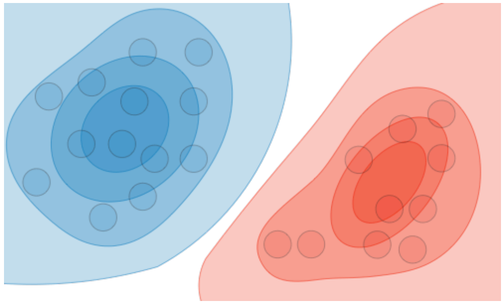
E.g., logistics/softmax regression

- **Generative learning**

$$P(Y = k\,|\vec{X} = \vec{x}) = \frac{P(\vec{X} = \vec{x}|Y = k)P(Y = k)}{P(\vec{X} = \vec{x})}$$

for $k = 0,1\ \ or\ 1,2, \ldots, K$

E.g., LDA/QDA



➢ Naive Bayes method - Generative learning

Maximum Likelihood:

$$P(\text{data}) = P(\boldsymbol{X}, \vec{\boldsymbol{y}}) = \prod_{i=1}^{n} P\left(\vec{X} = \vec{x}^{(i)}, Y = y^{(i)}\right)$$

$$= \prod_{i=1}^{n} P\left(\vec{X} = \vec{x}^{(i)} \mid Y = y^{(i)}\right) P(Y = y^{(i)})$$

Recall

$$P\left(\vec{X} \mid Y\right) = P(X_1, \dots X_d \mid Y)$$

$$= P(X_1|Y)P(X_2|X_1, Y)P(X_3|X_2, X_1, Y) \dots P(X_d|X_{d-1}, \dots, X_1, Y)$$

- **Independence**

A and B are **independent** if and only if $P(A \cap B) = P(A)P(B)$

if and only if $P(A|B) = P(A)$

if and only if $P(B|A) = P(B)$

- **Conditional Independence**

A and B are **conditional independent**

if and only if $P(A \cap B \mid C) = P(A|C)P(B|C)$

Remark: there is no direct relation between the above two.

Conditional Independence Joke:

A survey has pointed out a positive and significant correlation between the number of accidents and wearing heavy coats in Boston. They concluded that coats could hinder movements of drivers and be the cause of accidents. A new law was prepared to prohibit drivers from wearing coats when driving.

Finally, another study pointed out that people wear coats when it snows…

$$P(\text{Accident} \mid \text{Coats, Snow}) = P(\text{Accident} \mid \text{Snow})$$

$$P(\text{Accident, Coats} \mid \text{Snow}) = P(\text{Accident} \mid \text{Coats, Snow})P(\text{Coats} \mid \text{Snow})$$

$$= P(\text{Accident} \mid \text{Snow})P(\text{Coats} \mid \text{Snow})$$

Accident and coats are not independent, but they are conditional independent.

Naive Bayes Assumption:

Assume that the $\boldsymbol{X_i|Y}$ are conditionally independent given $\boldsymbol{Y}$.

Naive Bayes method Hypothesis for the model:

- $Y \backsim \text{Bernouli}\ (\phi)$ or $Y \backsim \text{Categorical}(\phi_1, \dots, \phi_K)$
- $\left(X_j \middle| Y = k\right) \backsim \text{Bernouli}\ \left(\phi_{j,k}\right)$

for $j = 1,2, \dots, d$, and $k = 0,1\ \ or\ 1,2, \dots, K$

PDF functions:

$$P(Y = y) = p_Y(y) = \phi_1^{\mathbb{I}(\mathrm{y=1})}\phi_2^{\mathbb{I}(\mathrm{y=2})} \dots \phi_K^{\mathbb{I}(\mathrm{y=K})}$$

$$= \prod_{k=1}^{K} (\phi_k)^{\mathbb{I}(y\ =k)}$$

For $k = 1,2, \dots, K$,

For $j = 1,2, \dots, d$,

$$P\left(X_j = x_j \middle| Y = k\right) = \left(\phi_{j,k}\right)^{x_j}\left(1 - \phi_{j,k}\right)^{1-x_j}$$

$$= \left(\phi_{j,k}\right)^{\mathbb{I}(x_j=1)}\left(1 - \phi_{j,k}\right)^{\mathbb{I}(x_j=0)}$$

Under **Naive Bayes assumption**, we **maximize likelihood**

$$L\left(\phi_{i,k}, \phi_k\right) = P(\boldsymbol{X}, \vec{\boldsymbol{y}}) = \prod_{i=1}^{n} P\left(\vec{X} = \vec{x}^{(i)} \mid Y = y^{(i)}\right) P(Y = y^{(i)})$$

$$= \prod_{i=1}^{n} P\left(X_1 = x_1^{(i)} \;\middle|\; Y = y^{(i)}\right) \cdots P\left(X_{\mathrm{d}} = x_{\mathrm{d}}^{(i)} \;\middle|\; Y = y^{(i)}\right) P\left(Y = y^{(i)}\right)$$

$$= \prod_{i=1}^{n} \prod_{j=1}^{d} P\left(X_{\mathrm{j}} = x_{\mathrm{j}}^{(i)} \;\middle|\; Y = y^{(i)}\right) P\left(Y = y^{(i)}\right)$$

$$= \prod_{i=1}^{n} \prod_{j=1}^{d} \prod_{k=1}^{K} \left(\phi_{j,k}\right)^{\mathbb{1}(x_j^{(i)}=1;\, y^{(i)}=k)} \left(1 - \phi_{j,k}\right)^{\mathbb{1}(x_j^{(i)}=0;\, y^{(i)}=k)} \left(\phi_k\right)^{\mathbb{1}(y^{(i)}=k)}$$

Equivalently, we **maximize log likelihood**

$$l(\phi_{j,k}, \phi_k) := \log L(\phi_{j,k}, \phi_k)$$

Calculate $\nabla\, l(\phi_{j,k}, \phi_k) = 0$ and find critical points.

We obtain formulas for the parameters maximizing the likelihood:

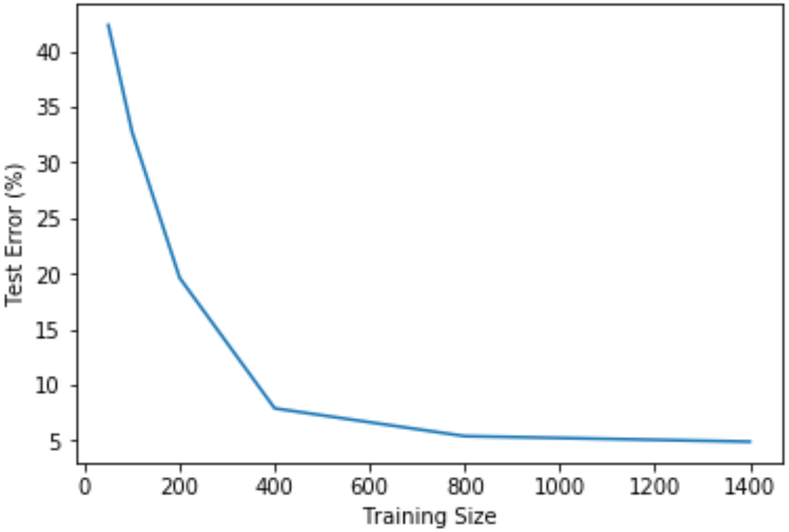
For $k = 1,2, \ldots, K$, and $j = 1,2, \ldots, d$,

$$\phi_k = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(y^{(i)} = k)$$

$$\phi_{j,k} = \frac{\sum_{i=1}^{n} \mathbb{I}(x_j^{(i)} = 1, y^{(i)} = k)}{\sum_{i=1}^{n} \mathbb{I}(y^{(i)} = k)}$$

Spam Email detection:

Data cleaning, vectorization,



Laplace Smoothing

Problem: For Spam/NonSpam example, if some word (e.g., the 1800-th dictionary word “XN-Project” is **not** in the training example.

Then, $P(X_{1800}\ |Y = k) = \phi_{1800,k} = 0$ for $k = 0,1$

Intuition Reason: Suppose someone did not pass the driving test for the 1st time, 2ed time, 3rd time, 4th time, 5th time, 6th time, 7th time, 8th time. What is the estimate pass rate for this student driver?

$$\phi = \frac{\sum_{i=1}^{n} \mathbb{I}(y^{(i)} = 1)}{n} = \frac{0}{8}$$

Calculation Reason:

$$P(Y = 1\ |\vec{X} = \vec{x}) = \frac{P(Y = 1,\ \vec{X} = \vec{x})}{P(\vec{X} = \vec{x})}$$

$$= \frac{\prod_{i=1}^{p} P(X_i = x_i|Y = 0)P(Y = 1)}{\prod_{i=1}^{p} P(X_i = x_i|Y = 1)P(Y = 1) + \prod_{i=1}^{p} P(X_i = x_i|Y = 0)P(Y = 0)} = \frac{0}{0}$$

Laplace Smoothing to the estimates:

For $k = 1,2, \dots, K$, and $j = 1,2, \dots, d$,

$$\phi_k = \frac{1 + \sum_{i=1}^{n} \mathbb{1}(y^{(i)} = k)}{n + K}$$

$$\phi_{j,k} = \frac{1 + \sum_{i=1}^{n} \mathbb{1}(x_j^{(i)} = 1, y^{(i)} = k)}{2 + \sum_{i=1}^{n} \mathbb{1}(y^{(i)} = k)}$$

- Under certain conditions, the Laplace smoothing actually gives the optimal estimator of ϕ_k.
- In practice, we don’t have to apply Laplace smoothing to ϕ_k. (Reason?)

Remarks:

1. Any assumptions may not be satisfied. This may also happen for Naive Bayes assumption.

2. Naive Bayes algorithm mainly for the case of problems where the features X_i are binary-valued.

3. What is the Naive Bayes Assumption for Gaussian Discriminant Analysis?

$$\vec{X} \mid Y = k \ \backsim\ \mathrm{Normal}(\vec{\mu}_k, \Sigma_\mathrm{k})$$

Another way to vectorize the data: (Event model)

Let $V = \{\vec{v}\}$. Here, $\vec{v}$ is the set of words of an email.

Let dictionary is a **basis B** for V.

Consider the coordinate map relative the dictionary basis:

$$V \xrightarrow{\hspace{5em}} \mathbb{Z}_d^m$$

Here, m is the email length and d is the dictionary size.

$\vec{v}$ ={Dear Good Friend, … … } $\xrightarrow{\hspace{5em}}$ $\vec{x} = \begin{bmatrix} 231 \\ 1086 \\ 349 \\ \vdots \\ \vdots \end{bmatrix}$

Naive Bayes method Hypothesis for the model:

- $Y \backsim \text{Categorical}(\phi_1, \dots, \phi_K)$

- $(X_j | Y = k) \backsim \text{Categorical}(\phi_{j,k,1}, \dots, \phi_{j,k,d})$

for $j = 1,2, \dots, d$, and $k = 0,1 \ \ or\ 1,2, \dots, K$

Furthermore, **assume**

$\phi_{j,k,l} = \phi_{j',k,l}$ for all j and j' (denoted by $\phi_{k,l}$)

A word is generated does not depend on its position j within the email.

Under **Naive Bayes assumption**, we **maximize likelihood**

$$L(\phi_{k,l}, \phi_k) = P(\boldsymbol{X}, \vec{\boldsymbol{y}}) = \prod_{i=1}^{n} P\left(\vec{X} = \vec{x}^{(i)} \mid Y = y^{(i)}\right) P(Y = y^{(i)})$$

$$= \prod_{i=1}^{n} \prod_{j=1}^{m_i} P\left(X_{\mathrm{j}} = x_{\mathrm{j}}^{(i)} \;\middle|\; Y = y^{(i)}\right) P\left(Y = y^{(i)}\right)$$

$$= \prod_{i=1}^{n} \prod_{j=1}^{m_i} \prod_{l=1}^{d} \prod_{k=1}^{K} \left(\phi_{k,l}\right)^{\mathbb{I}(x_j^{(i)} = l;\, y^{(i)} = k)} \left(\phi_k\right)^{\mathbb{I}(y^{(i)} = k)}$$

Equivalently, we **maximize log likelihood**

$$l(\phi_{k,l}, \phi_k) := \log L(\phi_{k,l}, \phi_k)$$

Calculate $\nabla\, l(\phi_{k,l}, \phi_k) = 0$ and find critical points.

We obtain formulas for the parameters maximizing the likelihood:

For $k = 1,2, \ldots, K$, and $j = 1,2, \ldots, d$,

$$\phi_k = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(y^{(i)} = k)$$

$$\phi_{k,l} = \frac{\sum_{i=1}^{n} \sum_{j}^{m_i} \mathbb{1}(x_j^{(i)} = l, y^{(i)} = k)}{\sum_{i=1}^{n} \mathbb{1}(y^{(i)} = k)}$$

We can also apply Laplace smoothing for the above estimates.

Remarks:

If we count the number of times words appeared, we need to use binomial distribution or multinomial distribution.

- **Binomial** is a generalization of Bernoulli distribution.

Given a series of n independent trials with two outcomes (T or F) with constant probability p and $1-p$.

Let X be the number of T appears in the n trials. Then $\mathrm{X} \backsim Binomial(n, p)$

$$P(X=k)=\binom{n}{k}p^k(1-p)^{n-k}$$

For example, flip a coin n times.

- **Multinomial** is a generalization of Categorical distribution.

Given a series of n independent trials with m outcomes $(O_1, \ldots O_m)$ with constant probability $(\phi_1, \ldots ,\ \phi_m)$.

Let $\vec{X}$ be the number of O_i appears in the n trials.

Then $\vec{X} \backsim Multinomial(n, \phi_1, \ldots ,\ \phi_m)$

$$P(X_i = n_i) = \frac{n!}{n_1! \cdots n_m!} \phi_1^{n_1} \cdots \ \phi_m^{n_m}$$

for each $i = 1, \ldots, m$, and each $n_1 + \cdots + n_m = n$

For example, Toss a K-side die n times.

More applications of Naive Bayes algorithm:

Naive Bayes algorithm mainly for the case of problems where the features are binary-valued (0,1) or multiclass valued (1,2,…,K).

For continuous value feature, we can discretize it to be multiclass and then apply Naive Bayes algorithm.

For example, for the feature house size, we might discretize the continuous values as follows:

House size (1000 sq feet)	<1	1-1.3	1.3-1.6	1.6-1.9	1.9-2.2	>2.2
new feature	1	2	3	4	5	6