



Universitat
de les Illes Balears

Artificial Intelligence

Lesson 4.4: Generative models

Naïve Bayes, Gaussian Discriminant Analysis



Discriminative models

Models used in machine learning for modeling the dependence of target variables y (usually hidden) on observed variables \mathbf{x} .

Within a probabilistic framework, this is done by modeling the conditional probability distribution

$$p(y|\mathbf{x})$$

For example:

The logistic regression model

$$h_{\theta}(\mathbf{x}) = g\left(\sum_{j=0}^n \theta_j x_j\right)$$

where

$$g(z) = \frac{1}{1 + e^{-z}}$$

Generative models

Models of the conditional probability of the observable \mathbf{x} , given a target variable y .

Within a probabilistic framework, this is done by modeling the conditional probability distribution

$$p(\mathbf{x}|y)$$

...and

$$p(y)$$

For example:

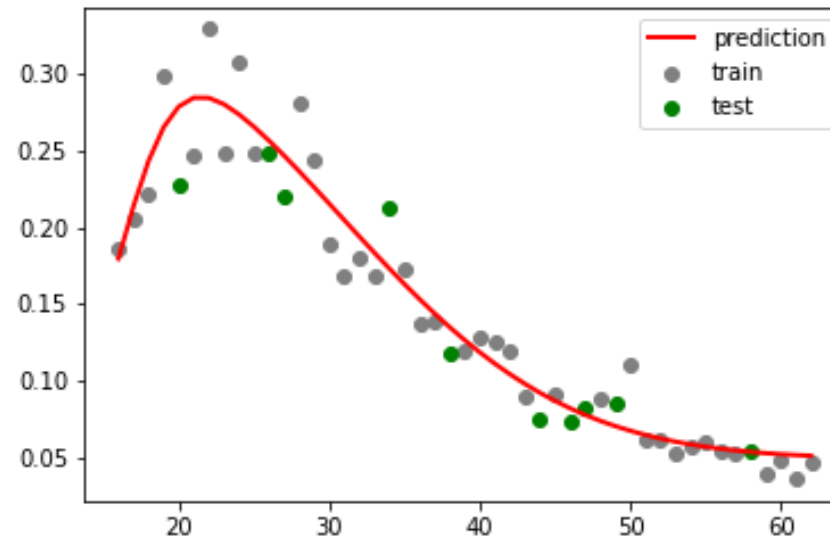


$$p(\mathbf{x}|y = \text{cat})$$

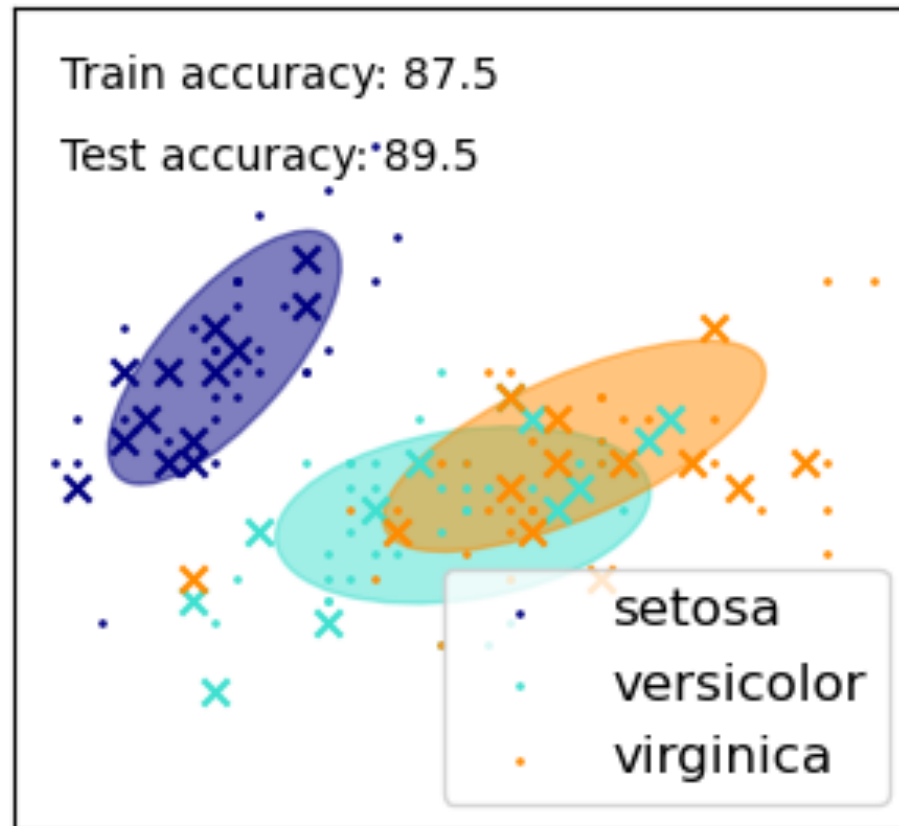


$$p(\mathbf{x}|y = \text{dog})$$

Gaussian for function approximation



Gaussian for classification



The multivariate Gaussian distribution

The multivariate normal distribution in n-dimensions, also called the multivariate **Gaussian** distribution, also written $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$, is given by

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

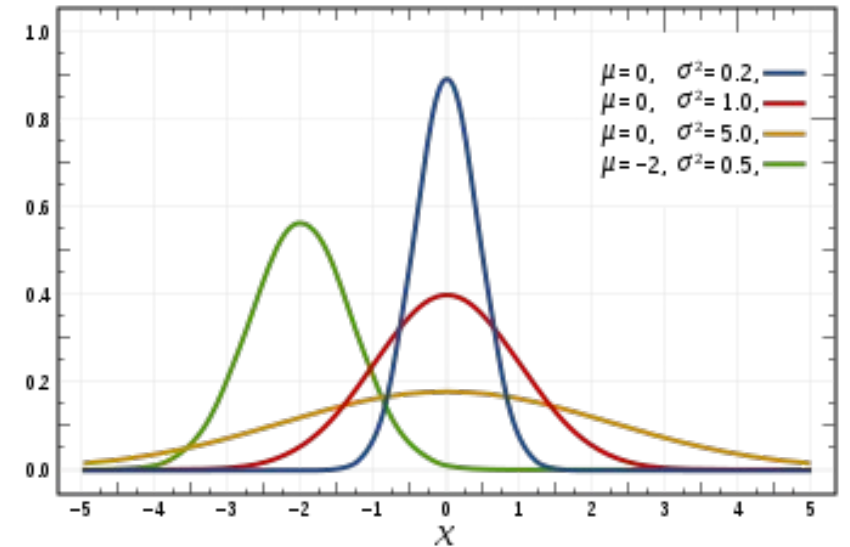
that it is parametrized by a **mean** vector $\boldsymbol{\mu} \in \mathbb{R}^n$ and a **covariance** matrix $\Sigma \in \mathbb{R}^n \times \mathbb{R}^n$.

The multivariate Gaussian (1D)

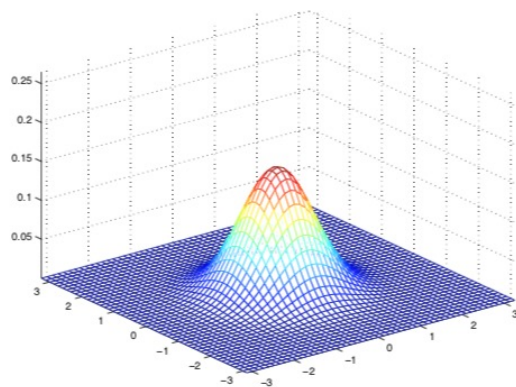
The 1D **Gaussian** distribution, also written $\mathcal{N}(\mu, \Sigma)$, is given by

$$p(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} (x - \mu)^2\right)$$

that it is parametrized by a **mean** $\mu \in \mathbb{R}$ and a **variance** $\sigma^2 \in \mathbb{R}$.

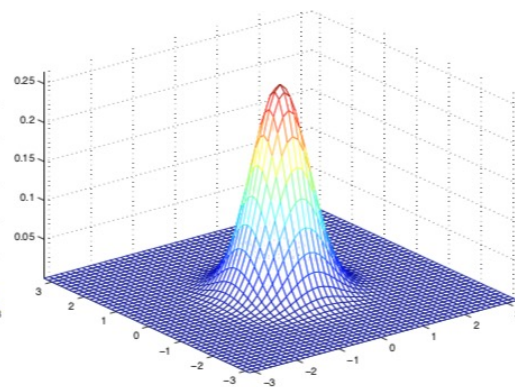


The multivariate Gaussian (2D)



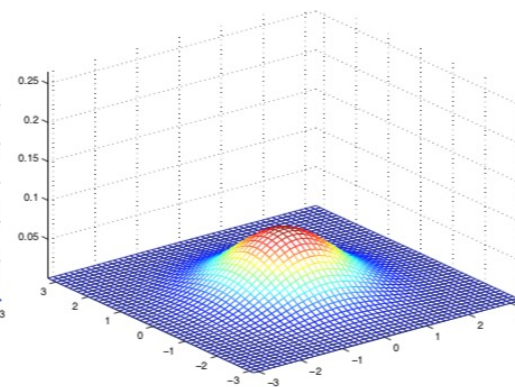
$$\boldsymbol{\mu} = (0,0)^T$$

$$\Sigma = I$$



$$\boldsymbol{\mu} = (0,0)^T$$

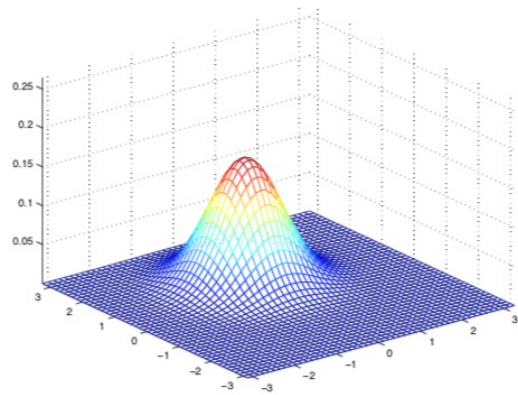
$$\Sigma = 0.6I$$



$$\boldsymbol{\mu} = (0,0)^T$$

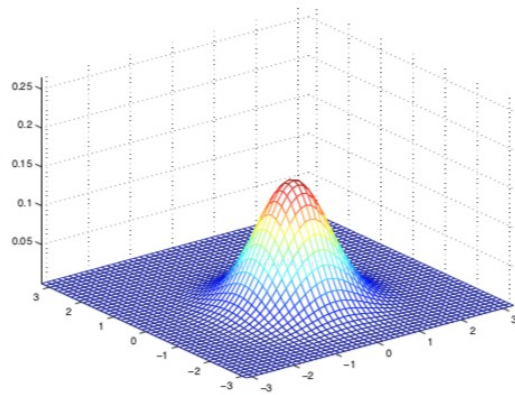
$$\Sigma = 2I$$

The multivariate Gaussian (2D)



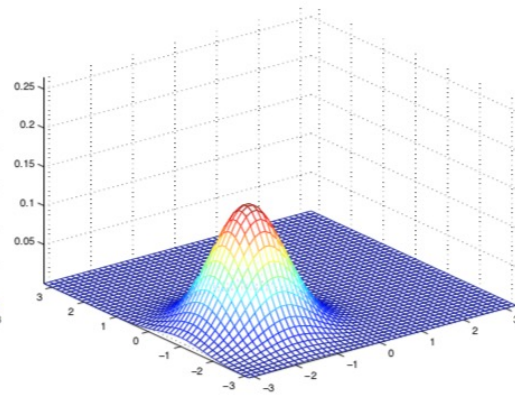
$$\boldsymbol{\mu} = (0, 1)^T$$

$$\boldsymbol{\Sigma} = I$$



$$\boldsymbol{\mu} = (0, -0.5)^T$$

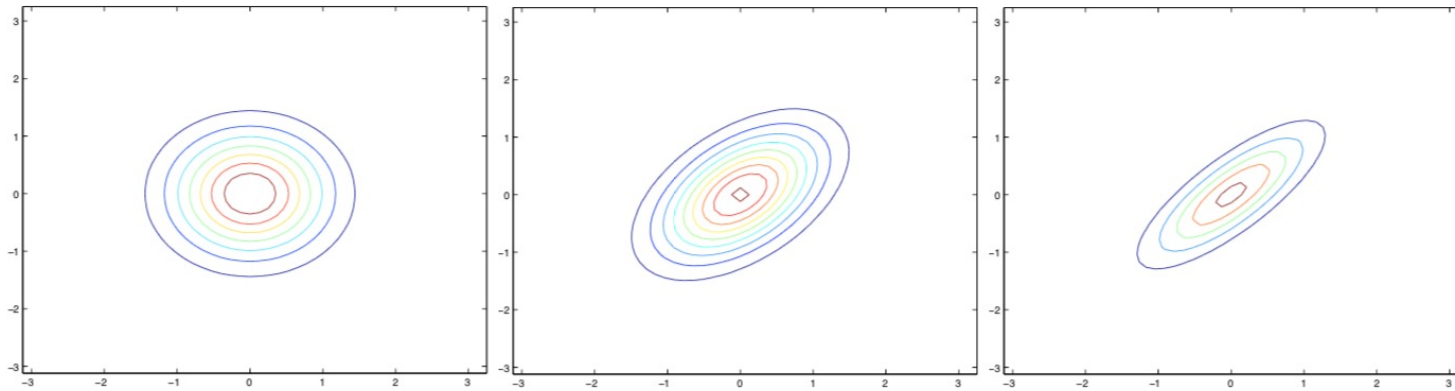
$$\boldsymbol{\Sigma} = I$$



$$\boldsymbol{\mu} = (1, -1.5)^T$$

$$\boldsymbol{\Sigma} = I$$

The multivariate Gaussian (2D)



$$\boldsymbol{\mu} = (0,0)^T$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\boldsymbol{\mu} = (0,0)^T$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

$$\boldsymbol{\mu} = (0,0)^T$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$$

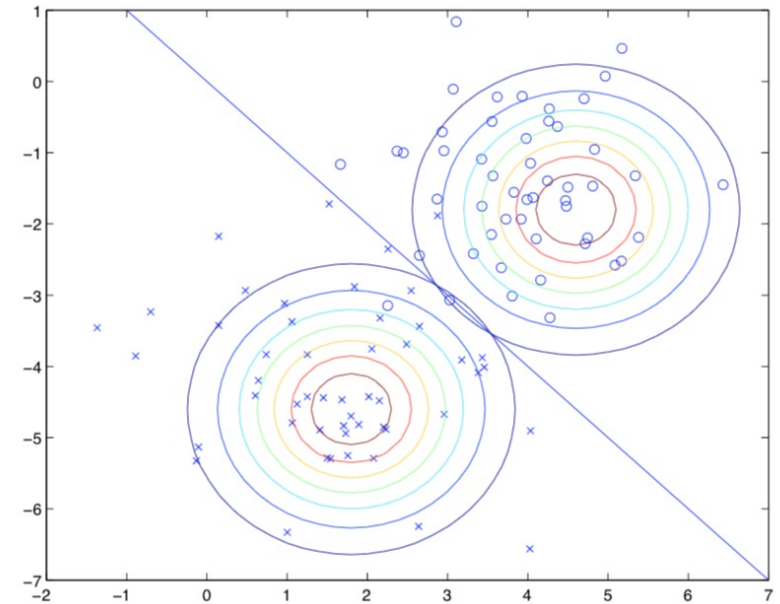
Gaussian Discriminant Analysis

Hypothesis:

GDA models $p(\mathbf{x}|y)$ using a multivariate normal distribution.

Therefore, the model is

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



Discussion: GDA and logistic regression

- if $p(\mathbf{x}|y)$ is a multivariate gaussian (with shared Σ), then $p(y|\mathbf{x})$ necessarily follows a logistic function.
- The converse, however, is not true.
- GDA makes stronger modelling assumptions, and is more data efficient (i.e., requires less training data to learn “well”) when the modelling assumptions are correct or at least approximately correct.
- Logistic regression makes weaker assumptions, and is significantly more robust to deviations from modelling assumptions.

Naïve Bayes

In GDA, the feature vectors \mathbf{x} were continuous, real-valued vectors. Let's now talk about a different learning algorithm in which the x_j 's are **discrete-valued**.

When the original, continuous-valued attributes are not well-modeled by a multivariate normal distribution, discretizing the features and using **Naïve Bayes** (instead of GDA) will often result in a better classifier.

m^2	<40	40-80	80-120	>120
x_j	1	2	3	4

The Naïve Bayes model

Hypothesis: Conditional Independence

Assume that the feature probabilities $p(x_j|y)$ are independent

$$p(\mathbf{x}|y = c) = \prod_{j=1}^n p(x_j|y = c)$$

given the class c .

The Naïve Bayes model derivation

$$h_{\theta}(\mathbf{x}) = \operatorname{argmax}_{c \in \mathcal{C}} p(y = c | \mathbf{x})$$

Bayes' Theorem

$$h_{\theta}(\mathbf{x}) = \operatorname{argmax}_{c \in \mathcal{C}} \frac{p(\mathbf{x} | y = c) p(y = c)}{p(\mathbf{x})}$$

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

$$h_{\theta}(\mathbf{x}) = \operatorname{argmax}_{c \in \mathcal{C}} p(\mathbf{x} | y = c) p(y = c)$$

Conditional
independence

$$p(\mathbf{x} | y = c) = \prod_{j=1}^n p(x_j | y = c)$$

$$h_{\theta}(\mathbf{x}) = \operatorname{argmax}_{c \in \mathcal{C}} \left(\prod_{j=1}^n p(x_j | y = c) \right) p(y = c)$$

Applications: NLP (natural lang. proc.)

- Spam detection
- Authorship identification
- Age/gender identification
- Language identification
- Assigning subject categories, topics, or genres
- Sentiment analysis

Subject: **Important notice!**
From: Stanford University <newsforum@stanford.edu>
Date: October 28, 2011 12:34:16 PM PDT
To: undisclosed-recipients;

Greats News!

You can now access the latest news by using the link below to login to Stanford University News Forum.

<http://www.123contactform.com/contact-form-StanfordNew1-236335.html>

Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information about the new services.

© Stanford University. All Rights Reserved.



Example: Text Classification

Definitions

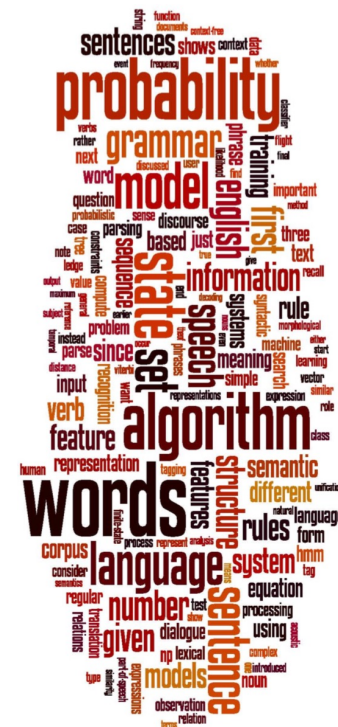
Input:

- a document/s (d)
- a fixed set of classes

$$C = \{c_1, c_2, \dots\}$$

Output:

- a predicted class $c \in \mathcal{C}$



A Spam Filter

We will represent an email via a feature vector whose length is equal to the number of words in the dictionary. Specifically, if an email contains the j -th word of the dictionary, then we will set $x_j = 1$; otherwise, we let $x_j = 0$. For instance, the next vector...

$$x = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \quad \begin{matrix} a \\ \text{aardwark} \\ \text{aarwolf} \\ \vdots \\ \text{buy} \\ \vdots \\ \text{zygmurgy} \end{matrix} \quad \left. \vphantom{\begin{matrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{matrix}} \right\} \text{Vocabulary} \rightarrow \text{Problem: usually } n \text{ is very big...}$$

...is used to represent an email that contains the words "a" and "buy," but not "aardvark," "aardwolf" or "zygmurgy."

A Spam Filter

Actually, rather than looking through an English dictionary for the list of all English words, in practice it is more common to look through our training set and encode in our feature vector only the words that occur at least once there.

Apart from reducing the number of words modeled and hence reducing our computational and space requirements, this also has the advantage of allowing us to model/include as a feature many words that may appear in your email (such as “deeplearning”) but that you won’t find in a dictionary.

Sometimes, we also exclude the very high frequency words (which will be words like “the,” “of,” “and”; these high frequency, “content free” words are called **stop words**) since they occur in so many documents and do little to indicate whether an email is spam or non-spam.

A Spam Filter

If $y = 1$ means spam email; “buy” is word 2087 and “price” is word 39831; then we are assuming that if I tell you $y = 1$ (that a particular piece of email is spam), then knowledge of x_{2087} (knowledge of whether “buy” appears in the message) will have no effect on your beliefs about the value of x_{39831} (whether “price” appears). More formally, we are only assuming that x_{2087} and x_{39831} are conditionally independent given y .)

$$p(x_1, \dots, x_{50000} | y) = p(x_1 | y) p(x_2 | y) \cdots p(x_{50000} | y) = \prod_{j=1}^n p(x_j | y)$$

The Naïve Bayes learning (I)

1st approach: using **maximum likelihood** for binary inputs $x_j = \{0,1\}$

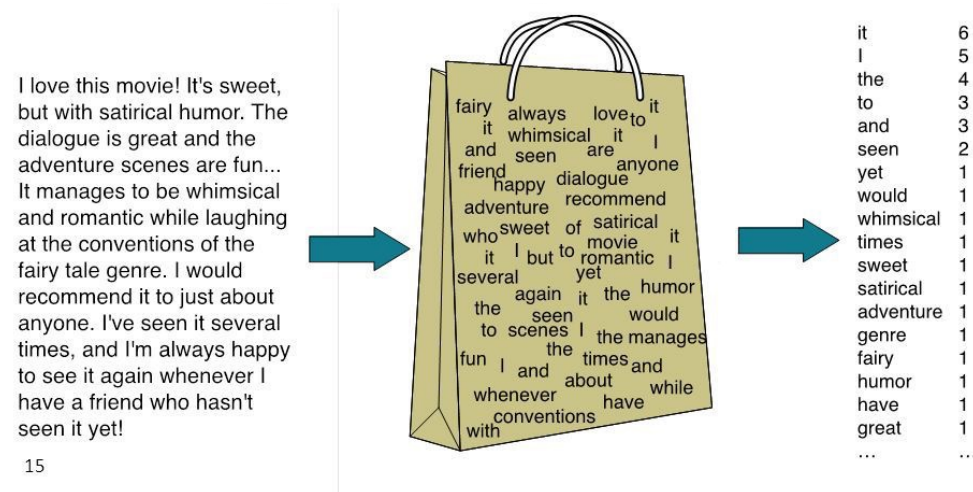
$$\theta_{j|y=c} = p(x_j|y=c) = \frac{\sum_{i=1}^m 1\{x_j^{(i)} = 1 \cap y^{(i)} = c\}}{\sum_{i=1}^m 1\{y^{(i)} = c\}}$$

$$\theta_{y=c} = p(y=c) = \frac{\sum_{i=1}^m 1\{y^{(i)} = c\}}{m}$$

Ex: $\theta_{j|y=1}$ is just the fraction of the spam ($y = 1$) emails in which word j does appear.

Bag of words

Relies on very simple representation of a document



15

The document d is represented as features $\mathbf{x} = (x_1, \dots, x_n)$ where

x_j = counts of word j in d

The Naïve Bayes learning (II)

2nd approach: using **maximum likelihood** for discrete inputs x_j

$$\theta_{j|y=c} = p(x_j|y=c) = \frac{\text{count}(x_j, y=c)}{\sum_{i=1}^m \text{count}(x_i, y=c)}$$

$$\theta_{y=c} = p(y=c) = \frac{\text{count}(y=c)}{m}$$

Naïve Bayes example

“Will the players play if weather is sunny?”

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

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

$$p(y = \text{Yes}|x = \text{Sunny}) = \frac{p(x = \text{sunny}|y = \text{Yes})p(y = \text{Yes})}{p(x = \text{Sunny})}$$

$$p(y = \text{No}|x = \text{Sunny}) = \frac{p(x = \text{sunny}|y = \text{No})p(y = \text{No})}{p(x = \text{No})}$$

Naïve Bayes example

$$\theta_{j|y=c} = p(x_j|y=c) = \frac{\text{count}(x_j, y=c)}{\sum_{i=1}^m \text{count}(x_i, y=c)}$$

$$\theta_{y=c} = p(y=c) = \frac{\text{count}(y=c)}{m}$$

“Will the players play if weather is sunny?”

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

$$p(y|x = \text{Sunny}) = \max_c p(x = \text{Sunny} | y = c)$$

$$p(x = \text{sunny} | y = \text{Yes})p(y = \text{Yes}) ?$$

$$p(x = \text{sunny} | y = \text{No})p(y = \text{No}) ?$$

$$\left. \begin{array}{l} p(x = \text{sunny} | y = \text{No}) = \frac{2}{5} \\ p(y = \text{No}) = \frac{5}{14} \end{array} \right\} \frac{2}{5} \cdot \frac{5}{14} = 0,14$$

Naïve Bayes example

$$\theta_{j|y=c} = p(x_j|y=c) = \frac{\text{count}(x_j, y=c)}{\sum_{i=1}^m \text{count}(x_i, y=c)}$$

$$\theta_{y=c} = p(y=c) = \frac{\text{count}(y=c)}{m}$$

“Will the players play if weather is sunny?”

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

$$p(y|x = \text{Sunny}) = \max_c p(x = \text{Sunny} | y = c)$$

$$p(x = \text{sunny} | y = \text{Yes})p(y = \text{Yes}) ?$$

$$p(x = \text{sunny} | y = \text{Yes}) = \frac{3}{9}$$

$$p(y = \text{Yes}) = \frac{9}{14}$$

$$\frac{3}{9} \cdot \frac{9}{14} = 0,21$$

Max!!

Prediction=Yes

Laplace smoothing for Naïve Bayes

Problem with maximum likelihood


$$\text{count}(x_j, y = c) = 0 \text{ ?}$$

Therefore

$$p(x_j|y = c) = 0 \quad p(y|x) = 0$$

Solution by means Laplace smoothing (add 1):

$$\theta_{j|y=c} = p(x_j|y = c) = \frac{\text{count}(x_j, y = c) + 1}{\sum_{i=1}^m \text{count}(x_i, y = c) + |V|}$$

“Vocabulary”


The Naïve Bayes learning (III)

3rd approach: using **maximum likelihood** for continuous inputs x_j , assuming Gaussian distributions

$$\theta_{j|y=c} = p(x_j|y=c) = \frac{1}{(2\pi\sigma_c)^{1/2}} \exp\left(-\frac{1}{2\sigma_c} (x_j - \mu_c)^2\right)$$

$$\theta_{y=c} = p(y=c) = \frac{\text{count}(y=c)}{m}$$

The Naïve Bayes model

While not necessarily the very best classification algorithm, the Naïve Bayes classifier often works surprisingly well. It is often also a very good “first thing to try,” given its simplicity and ease of implementation.

- Very Fast, low storage requirements
- Robust to Irrelevant Features
- Very good in domains with many equally important features
- Optimal if the independence assumptions hold
- A good dependable baseline for text classification

Exercise

	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?