Bayes Classifier

Yanyan Lan
lanyanyan@ict.ac.cn

Types of Classifiers

We can divide the large variety of classification approaches into three major types:

- Instance based classifiers
  - Use observation directly (no models)
  - *e.g.*, K Nearest Neighbors
- Discriminative classifiers
  - Directly estimate a decision rule/boundary
  - *e.g.*, Logistic Regression
- Generative classifiers
  - Build a generative statistical model
  - *e.g.*, Naïve Bayes

Frequentist vs. Bayesian

- 在 statistical inference 上，主要有两派：频率学派和贝叶斯学派
- Frequentist statistics tries to eliminate uncertainty by providing estimates.
- Bayesian statistics tries to preserve and refine uncertainty by adjusting individual beliefs in light of new evidence.
  - To produce quantitative trading strategies based on Bayesian models.

Frequentist vs. Bayesian

- 频率学派认为概率分布的参数是一个确定值，可以直接进行估计
  - 最大似然估计
- 贝叶斯学派认为并不能确定数据是用哪个固定参数造出来的，因此他们关心的是参数空间的每一个值，给这些值一些他们自己认为合理的假设值（先验分布），然后在去做实验（证据），不断地调整自己的假设，从而得到最后结果（后验分布）
  - 最大后验估计

Frequentist vs. Bayesian

- θ represents the parameters of the model
- Frequentist: it exists an true θ. For example, flip a coin 100 times and 20 times face up. So $\theta = \mathrm{P}(\text{head}) = \frac{20}{100} = 0.2$.
- Bayesian: θ is random variable which meets a certain probability distribution.

Bayesian

- Bayesian:
  - Prior: the knowledge before getting any data. for example, the coin has a high probability of being uniform, and a small probability is uneven.
  - Likelihood: $\mathrm{P}(X|\theta)$. The data we observe given θ
  - Posterior: $\mathrm{P}(\theta|X)$. The final parameter distribution.

$$\mathrm{P}(\theta|X) = \frac{\mathrm{P}(X|\theta)P(\theta)}{P(X)}$$

  - For example, flip a coin 5 times and 5 times face up, if we think the coin has a high probability of being uniform(eg. Prior is a Beta distribution which the maximum value is taken at 0.5). Then $\mathrm{P}(\theta|X)$ will be a distribution which the maximum value is taken between 0.5 and 1, instead of simply getting "θ = 1".

Frequentist vs. Bayesian

- Frequentist：
  - estimate → accurate when data → infinite
  - However, if there is a lack of data, serious deviations may occur.
  - eg. If we flip a coin 5 times and 5 times face up,

    frequentist estimates $\theta = 1$ while the true $\theta = 0.5$ for a uniform coin
- Bayesian:
  - has a prior distribution for θ to avoid the above situation
  - when data → larger

    the influence of data → larger and the prior → weaker.

Bayesian Decision Theory

Classification as Bayesian Decision

- Credit scoring example:
  - Inputs: income and savings, or $\mathbf{x} = (x_1, x_2)^\top$
  - Output: risk $\in \{\text{low}, \text{high}\}$, or $C \in \{0,1\}$
- Prediction:

$$\begin{cases} C = 1 & \text{if } p(c = 1|\mathbf{x}) > 0.5 \\ C = 0 & \text{otherwise} \end{cases}$$

or equivalently

$$\begin{cases} C = 1 & \text{if } p(c = 1|\mathbf{x}) > p(c = 0|\mathbf{x}) \\ C = 0 & \text{otherwise} \end{cases}$$

Losses and Risks

- Different decisions or actions may not be equally good or costly.
- Action α_{i}: decision to assign the input **x** to class C_i
- misclassification Loss λ_{ik}: loss incurred for taking action α_i when the actual state is C_k
- Expected risk for taking action α_i :

$$R(C_i|\mathbf{x}) = \sum_{k=1}^{K} \lambda_{ik}\, p(C_k|\mathbf{x})$$

- Optimal decision rule with minimum expected risk:

$$h^*(\mathbf{x}) = \arg\min_i R(C_i|\mathbf{x})$$

h^* is called Bayes Optimal Classifier, the corresponding risk is called Bayes risk.

0-1 Loss

- All correct decisions have no loss and all errors have unit cost:

$$\lambda_{ik} = \begin{cases} C = 0 & \text{if } i = k \\ C = 1 & \text{if } i \neq k \end{cases}$$

- Expected risk:

$$R(C_i|\boldsymbol{x}) = \sum_{k=1}^{K} \lambda_{ik}\, p(C_i|\boldsymbol{x})$$

- Optimal decision rule with minimum expected risk (or, equivalently, highest posterior probability):

$$h^*(\boldsymbol{x}) = \arg\max_{i} p(C_i|\boldsymbol{x})$$

Discriminative Models

- One way of performing classification is called discriminative functions.

- Discriminative models model p(c|**x**) directly, eg. Logistic regression…

- Different ways of defining the discriminant functions

$$g_i(\mathbf{x}) = -R(\alpha_i|\mathbf{x})$$

$$g_i(\mathbf{x}) = p(C_i|\mathbf{x})$$

Generative Models

- Another way of performing classification is called
  generative models
- Generative models model joint probability P(**x**, c) first, then get P(c|**x**)

$$P(c|\boldsymbol{x}) = \frac{P(\boldsymbol{x}, c)}{P(\boldsymbol{x})}$$

- By Bayes' theorem

$$P(c|\boldsymbol{x}) = \frac{P(c)P(\boldsymbol{x}|c)}{P(\boldsymbol{x})}$$

Where P(c) is called prior probability, P(c|**x**) poster probability.

P(**x**|c) is class-conditional probability or likelihood.

Bayes' Rule for $K > 2$ Classes

Naïve Bayes is a generative model.

If we know the conditional probability $p(\mathbf{x} | C_i)$ we can determine the appropriate class by using Bayes rule:

$$p(C_i|\mathbf{x}) = \frac{p(\mathbf{x}|C_i)p(Ci)}{p(\mathbf{x})} \propto p(\mathbf{x}|C_i)p(Ci)$$

Bayes' Rule for $K > 2$ Classes

If we know the conditional probability $p(\mathbf{x}|C_i)$ we can determine the appropriate class by using Bayes rule:

$$p(C_i|\mathbf{x}) = \frac{p(\mathbf{x}|C_i)p(Ci)}{p(\mathbf{x})} \propto p(\mathbf{x}|C_i)p(Ci)$$

But how do we determine $p(\mathbf{x}|C_i)$?

Naïve Bayes Classifier

- Naïve Bayes classifiers assume that given the class label $y = C_i$ the attributes are conditionally independent of each other.

$$p(\mathbf{x}|y = C_i) = \prod_{j=1}^{d} p(xj|y = Ci)$$

where x_j is the attribute for sample **x**, d is the dimension.

- Using this idea the full classification rule becomes:

$$\begin{aligned}\hat{y} &= \arg\max_k p(y = C_k|\mathbf{x}) \\ &= \arg\max_k \frac{p(C_k)p(\mathbf{x}|y = C_k)}{p(\mathbf{x})} \\ &= \arg\max_k p(C_k) \prod_{j=1}^{d} p(x_j|y = C_k)\end{aligned}$$

Likelihood

- Given a training data $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ the joint log likelihood of the data:

$$\mathcal{L}(\mathcal{D}) = \log p(\mathbf{X}, \mathbf{y}) = \log \prod_{i=1}^{N} \prod_{j=1}^{d} p(x_j^{(i)} | y^{(i)}) p(y^{(i)})$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{d} \left[\log p(x_j^{(i)} | y^{(i)}) + \log p(y^{(i)}) \right]$$

Estimation

$$\mathcal{L}(\mathcal{D}) = \sum_{i=1}^{N} \sum_{j=1}^{d} [\log p \left(x_j^{(i)} \middle| y^{(i)} \right) + \log p(y^{(i)})]$$

- Assume all attributes are binary (Bernoulli Naïve Bayes).
- To determine the MLE parameters for $p(x_j = 1|y = Ck)$, we simply count the times label C_k is seen in conjunction with x_j:

$$p(x_j = 1|y = C_k) = \frac{\sum_{i=1}^{N} \mathbb{1}\{x_j^{(i)} = 1 \cap y^{(i)} = C_k\}}{\sum_{i=1}^{N} \mathbb{1}\{y^{(i)} = C_k\}}$$

$$p(y = C_k) = \frac{\sum_{i=1}^{N} \mathbb{1}\{y^{(i)} = C_k\}}{N}$$

Prediction

- Once we computed all parameters for attributes in both classes we can easily decide on the label of a new sample **x**:

$$
\begin{aligned}
\hat{y} &= \arg\max_k p(y = C_k | \mathbf{x}) \\
&= \arg\max_k \frac{p(\mathbf{x} | y = C_k) p(y = C_k)}{p(\mathbf{x})} \\
&= \arg\max_k \prod_{j=1}^{d} p(x_j | y = C_k) p(C_k)
\end{aligned}
$$

Example: Text classification


boston.com

HOME OBITUARIES SPORTS ENTERTAINMENT BUSINESS LIFESTYLE HEALTH TRAVEL CARS JOBS REAL ESTA

NEWS SHAR

The story behind Mitt Romney's loss in the presidential campaign to President Obama

By Michael Kranish
Globe Staff
DECEMBER 22, 2012 7:00 PM

...who believe that they are entitled to health care, to food, to housing, to you-name-it. That that's an entitlement. And the government should give it to them. And they will vote for this president no matter what.

Mother Jones VIDEO

A video from a May fund-raiser in Florida showed Romney characterizing nearly half of Americans as "victims" who want
sociated Press photo of Mother Jones video) Credit: Associated Press photo of Mother Jones video

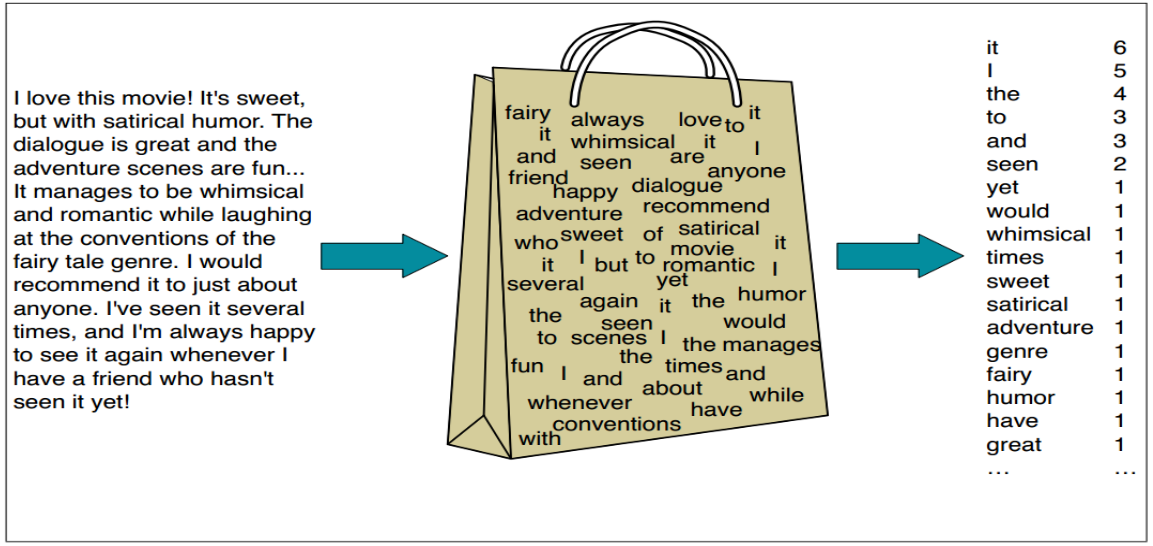
What is the major topic of this article?

Feature Transformation

- How do we encode the set of features (words) in the document?
- What type of information do we wish to represent? What can we ignore?

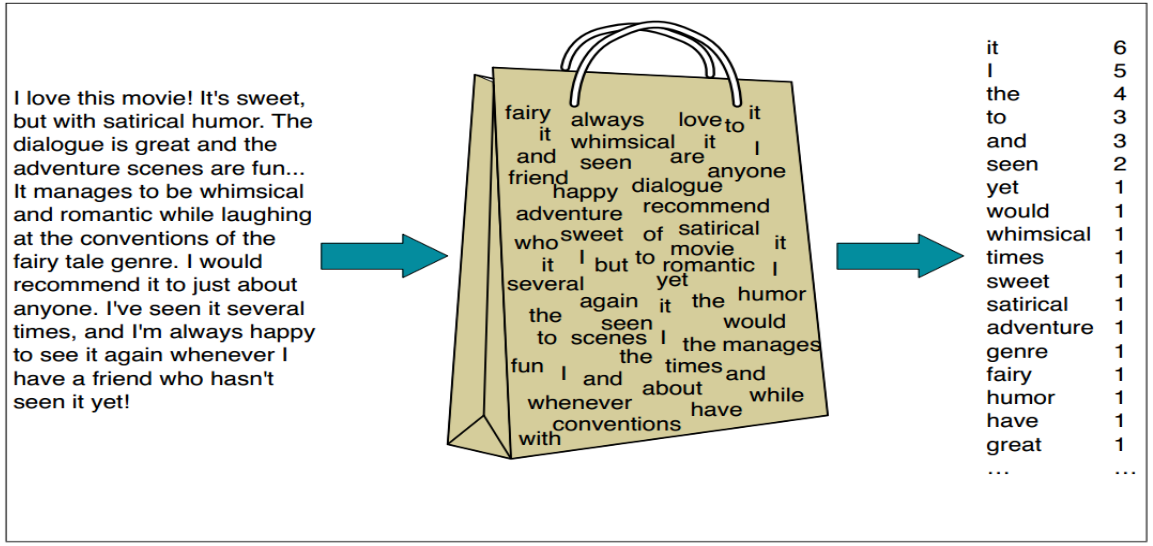
Feature Transformation

- How do we encode the set of features (words) in the document?
- What type of information do we wish to represent? What can we ignore?



Feature Transformation

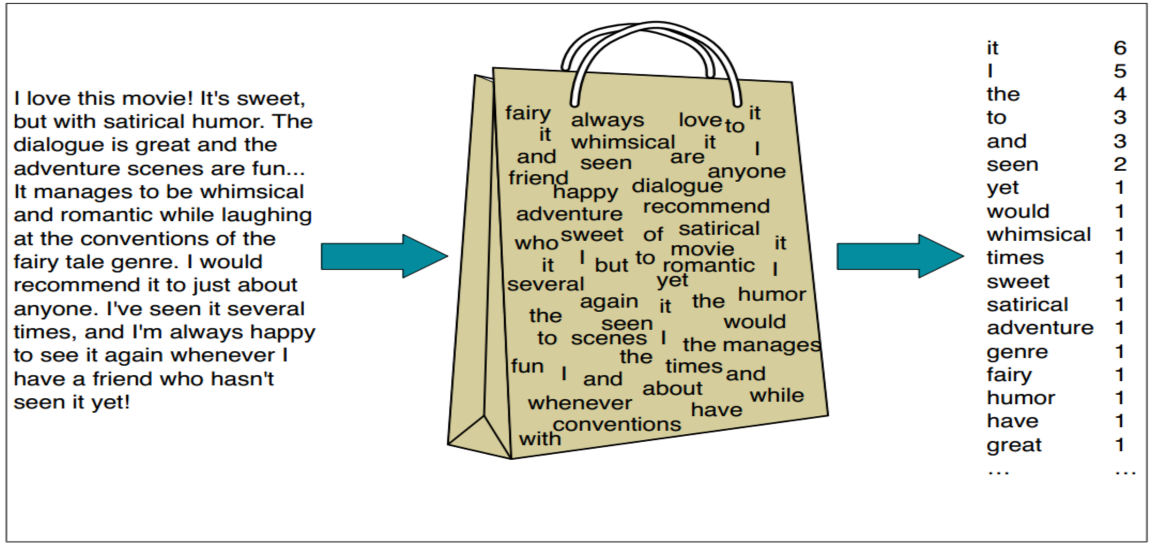
- How do we encode the set of features (words) in the document?
- What type of information do we wish to represent? What can we ignore?



- Most common encoding: “Bag of Words“

Feature Transformation

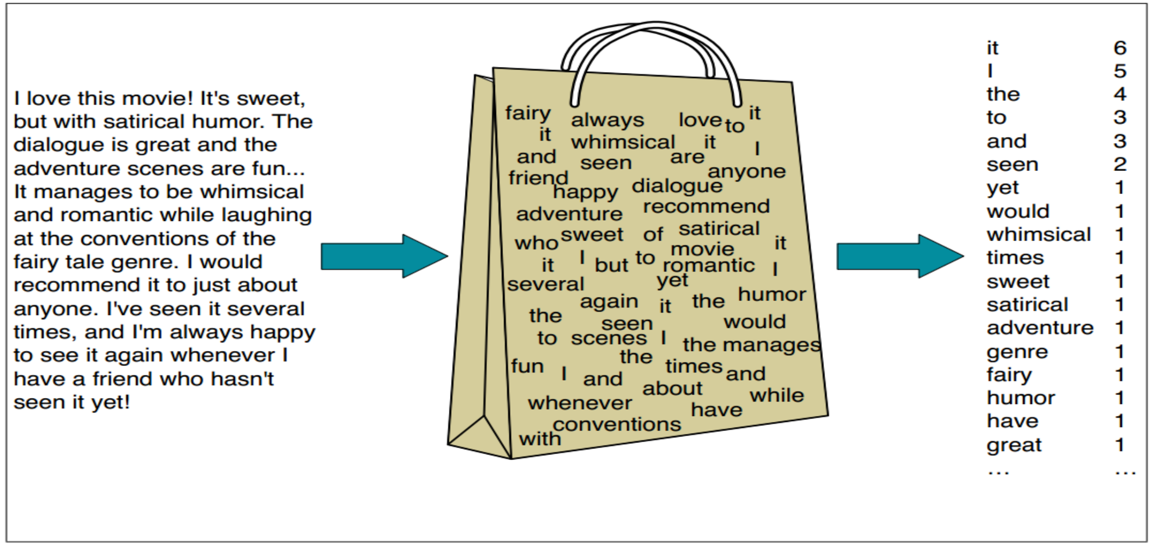
- How do we encode the set of features (words) in the document?
- What type of information do we wish to represent? What can we ignore?



- Most common encoding: “Bag of Words“
- Treat document as a collection of words and encode each document as a vector based on some dictionary.

Feature Transformation

- How do we encode the set of features (words) in the document?
- What type of information do we wish to represent? What can we ignore?



- Most common encoding: “Bag of Words“
- Treat document as a collection of words and encode each document as a vector based on some dictionary.
- The vector can either be binary or discrete.

Feature Transformation: Bag of Words

- In this example we will use a binary vector.
- For document $\mathbf{x}$ we will use a vector of d indicator features $\phi_j(\mathbf{x})$ for whether a word appears in the document:
  - $\phi_j(\mathbf{x}) = 1$: if word j appears in the document $\mathbf{x}$.
  - $\phi_j(\mathbf{x}) = 0$: if word j does not appear in the document $\mathbf{x}$.
- $\boldsymbol{\phi}(\mathbf{x}) = [\phi_1(\mathbf{x}), \ldots, \phi_d(\mathbf{x})]$ is the resulting feature vector for the entire dictionary for document $\mathbf{x}$.
- For notational simplicity we will replace each document $\mathbf{x}^{(i)}$ with a fixed length vector $\boldsymbol{\phi}_i = [\phi_1, \ldots, \phi_d]$, where $\phi_j = \phi_j(\mathbf{x}^{(i)})$.

Example

boston.com

HOME OBITUARIES SPORTS ENTERTAINMENT BUSINESS LIFESTYLE HEALTH TRAVEL CARS JOBS REAL ESTA

NEWS

SHAR

The story behind Mitt Romney's loss in the presidential campaign to President Obama

By Michael Kranish
Globe Staff
DECEMBER 22, 2012 7:00 PM

A video from a May fund-raiser in Florida showed Romney characterizing nearly half of Americans as "victims" who want
sociated Press photo of Mother Jones video) Credit: Associated Press photo of Mother Jones video

Dictionary

- Washington
- Congress

. . .

54. Romney
55. Obama
56. Nader

$$\phi_{54}(\mathbf{x}^i) = 1$$

$$\phi_{55}(\mathbf{x}^i) = 1$$

$$\phi_{56}(\mathbf{x}^i) = 0$$

Example: cont.

We would like to classify documents as election related or not.

- Given a collection of documents with their labels we learn the parameters for our model.
- For example, if we see the word "Obama" in n_1 out of the N documents labeled as "election" we set $p(\text{Obama}|\text{election}) = n_1/N$
- Similarly we compute the priors ($p(\text{election})$) based on the proportion of the documents from both classes.

Example: Classifying Election(E) or Sports(S)

- Assume we learned the following model

$$P(\phi_{romney} = 1|E) = 0.8, \quad P(\phi_{romney} = 1|S) = 0.1, \quad P(S) = 0.5$$
$$P(\phi_{obama} = 1|E) = 0.9, \quad P(\phi_{obama} = 1|S) = 0.05, \quad P(E) = 0.5$$
$$P(\phi_{clinton} = 1|E) = 0.9, \quad P(\phi_{clinton} = 1|S) = 0.05$$
$$P(\phi_{football} = 1|E) = 0.1, \quad P(\phi_{football} = 1|S) = 0.7$$

Example: Classifying Election(E) or Sports(S)

- Assume we learned the following model

$$P(\phi_{romney} = 1|E) = 0.8, \quad P(\phi_{romney} = 1|S) = 0.1, \quad P(S) = 0.5$$
$$P(\phi_{obama} = 1|E) = 0.9, \quad P(\phi_{obama} = 1|S) = 0.05, \quad P(E) = 0.5$$
$$P(\phi_{clinton} = 1|E) = 0.9, \quad P(\phi_{clinton} = 1|S) = 0.05$$
$$P(\phi_{football} = 1|E) = 0.1, \quad P(\phi_{football} = 1|S) = 0.7$$

- For a specific document we have the following feature vector:

$$\phi_{romney} = 1, \phi_{obama} = 1, \phi_{clinton} = 1, \phi_{football} = 0$$

Example: Classifying Election(E) or Sports(S)

- Assume we learned the following model

$$P(\phi_{romney} = 1|E) = 0.8, \quad P(\phi_{romney} = 1|S) = 0.1, \quad P(S) = 0.5$$
$$P(\phi_{obama} = 1|E) = 0.9, \quad P(\phi_{obama} = 1|S) = 0.05, \quad P(E) = 0.5$$
$$P(\phi_{clinton} = 1|E) = 0.9, \quad P(\phi_{clinton} = 1|S) = 0.05$$
$$P(\phi_{football} = 1|E) = 0.1, \quad P(\phi_{football} = 1|S) = 0.7$$

- For a specific document we have the following feature vector:

$$\phi_{romney} = 1, \phi_{obama} = 1, \phi_{clinton} = 1, \phi_{football} = 0$$

- Thus

$$p(E|1,1,1,0) \propto 0.8 * 0.9 * 0.9 * 0.9 * 0.5 = 0.2916$$
$$p(S|1,1,1,0) \propto 0.1 * 0.05 * 0.05 * 0.3 * 0.5 = 0.0000375$$

Smoothing

What if a document **x** contains word *j* which is missing in training set?

Smoothing

What if a document $\mathbf{x}$ contains word j which is missing in training set?

$$p(x_j|C_k) = \frac{\sum_{i=1}^{N} \mathbb{1}\{x_j^{(i)} = 1 \cap y^{(i)} = C_k\}}{\sum_{i=1}^{N} \mathbb{1}\{y^{(i)} = C_k\}} = 0$$

Thus:

$$p(C_k|\mathbf{x}) \propto 0, \quad \forall k$$

Smoothing

What if a document **x** contains word *j* which is missing in training set?

$$p(x_j|C_k) = \frac{\sum_{i=1}^{N} \mathbb{1}\{x_j^{(i)} = 1 \cap y^{(i)} = C_k\}}{\sum_{i=1}^{N} \mathbb{1}\{y^{(i)} = C_k\}} = 0$$

Thus:

$$p(C_k|\mathbf{x}) \propto 0, \quad \forall k$$

Solutions?

Laplace Smoothing

- To avoid this, we can use Laplace Smoothing, which replaces the above estimate with:

$$p(x_j|C_k) = \frac{\sum_{i=1}^{N} \mathbb{1}\{x_j^{(i)} = 1 \cap y^{(i)} = C_k\} + 1}{\sum_{i=1}^{N} \mathbb{1}\{y^{(i)} = C_k\} + K} = 0$$

where for binary classification $K = 2$.

Naïve Bayes Classifiers for Continuous Values

- So far we assumed a binomial or discrete distribution for the data given the model $p(x_j|y)$.
- However, in many cases the data contains continuous features:
  - Height, weight
  - Brain activity
  - . . .
- For these types of data we often use a Gaussian model.
- In this model we assume that the observed input vector **x** is generated from the following distribution:

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

Gaussian Bayes Classifier

- Assume i -th sample is generated from a Gaussian PDF that depends on the the class C_k
- Here we can also use the Naïve Bayes assumption: attributes are independent given the class label C_k
- In the Gaussian model this means that the covariance matrix becomes a diagonal matrix.
- Thus, we only need to learn the values for the variance term for each attribute under class k : $x_j \sim \mathcal{N}(\mu_{jk}, \sigma_{jk})$

$$p(\mathbf{x}|y = C_k) = \prod_{j=1}^{\mathrm{d}} \frac{1}{\sqrt{2\pi}\sigma_{jk}} \exp(-\frac{(x_j - \mu_{jk})^2}{2\sigma_{jk}^2})$$

Gaussian Bayes Classifier example

- Distinguish children from adults based on size
  - classes: {a,c}, attributes: height [cm], weight [kg]
  - training examples: $\{x_h^{(i)}, x_w^{(i)}, y^{(i)}\}$, 4 adults, 12 children
- Class probabilities: $p(y=a) = \frac{4}{4+12} = 0.25; p(y=c) = 0.75$
- Model for adults: estimates the parameters using MLE
  - $Height \sim \mathcal{N}(\mu_{h,a}, \Sigma_{h,a}^2)$
  - $Weight \sim \mathcal{N}(\mu_{w,a}, \Sigma_{w,a}^2)$

$$\mu_{h,a} = \frac{\sum_{i=1}^{N} x_h^{(i)} * \mathbb{1}\{y^{(i)} = a\}}{\sum_{i=1}^{N} \mathbb{1}\{y^{(i)} = a\}}$$

$$\sigma_{h,a} = \frac{\sum_{i=1}^{N} (x_h^{(i)} - \mu_{h,a})^2 * \mathbb{1}\{y^{(i)} = a\}}{\sum_{i=1}^{N} \mathbb{1}\{y^{(i)} = a\}}$$

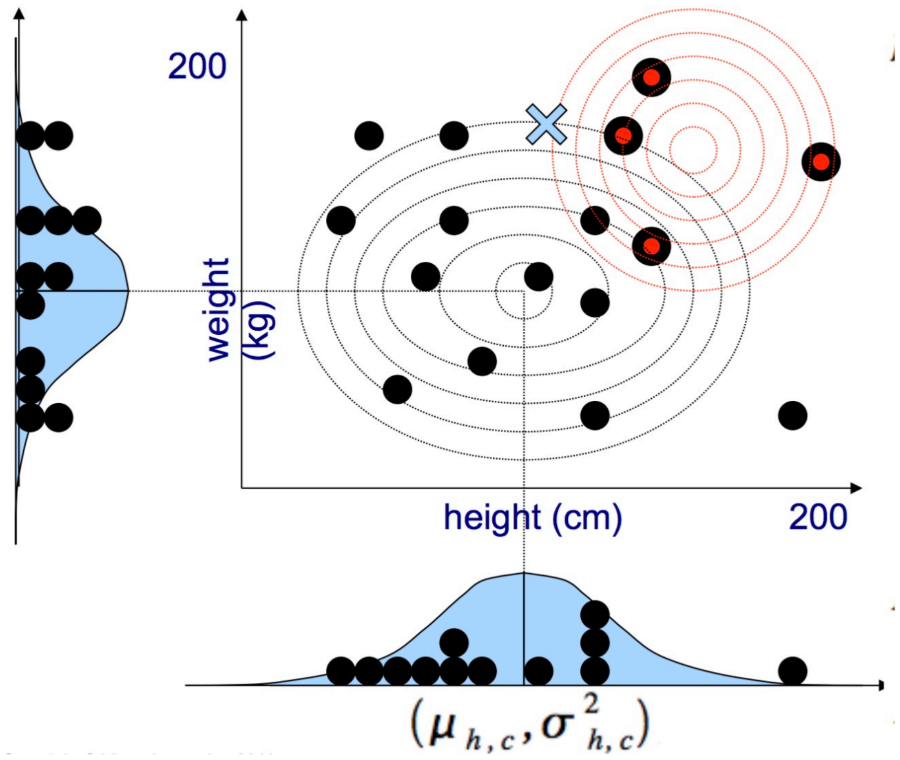
Gaussian Bayes Classifier example

$$\mu_{w,a} = \frac{\sum_{i=1}^{N} x_w^{(i)} * \mathbb{1}\{y^{(i)} = a\}}{\sum_{i=1}^{N} \mathbb{1}\{y^{(i)} = a\}}$$

$$\sigma_{w,a} = \frac{\sum_{i=1}^{N} (x_w^{(i)} - \mu_{w,a})^2 * \mathbb{1}\{y^{(i)} = a\}}{\sum_{i=1}^{N} \mathbb{1}\{y^{(i)} = a\}}$$

- Model for children: same way, estimating $(\mu_{h,c}, \sigma_{h,c}^2), (\mu_{w,c}, \sigma_{w,c}^2)$

Gaussian Bayes Classifier example



- assume height and weight independent

$$p(x_h|y=c) = \frac{1}{\sqrt{2\pi\sigma^2_{h,c}}} exp\{-\frac{1}{2}(\frac{(x_h-\mu_{h,c})^2}{\sigma^2_{h,c}})\}$$

$$p(x_w|y=c) = \frac{1}{\sqrt{2\pi\sigma^2_{w,c}}} exp\{-\frac{1}{2}(\frac{(x_w-\mu_{w,c})^2}{\sigma^2_{w,c}})\}$$

$$p(x_h|y=a) = \frac{1}{\sqrt{2\pi\sigma^2_{h,a}}} exp\{-\frac{1}{2}(\frac{(x_h-\mu_{h,a})^2}{\sigma^2_{h,a}})\}$$

$$p(x_w|y=a) = \frac{1}{\sqrt{2\pi\sigma^2_{w,a}}} exp\{-\frac{1}{2}(\frac{(x_w-\mu_{w,a})^2}{\sigma^2_{w,a}})\}$$

$$p(x|y=a) = p(x_h|y=a)p(x_w|y=a)$$

$$p(x|y=c) = p(x_h|y=c)p(x_w|y=c)$$

Bayes' theorem $\rightarrow$

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

Gaussian Bayes Classifier example

- For instance
  - given children data: (60, 60),(50, 50),(40, 40),(40, 40),(40, 40),(30, 30),(60, 60),(70, 70),(50, 50),(90, 90), (90, 90), (90, 90)
  - given adult data: (170, 170), (180, 180), (160, 160),(170,170)
  - estimate the parameters: By MLE

    $$\mu_{h,c} = 60, \sigma^2_{h,c} = 425, \mu_{w,c} = 60, \sigma^2_{w,c} = 425$$

    $$\mu_{h,a} = 170, \sigma^2_{h,a} = 50, \mu_{w,a} = 170, \sigma^2_{w,a} = 50$$
  - Now given the new data (120, 120), we need to predict it to adult or child

Gaussian Bayes Classifier example

$$p(x|y=a) = p(x_h|y=a)p(x_w|y=a)$$

$$= \frac{1}{\sqrt{2\pi\sigma_{h,a}^2}} exp\{-\frac{1}{2}(\frac{(x_h-\mu_{h,a})^2}{\sigma_{h,a}^2})\} \frac{1}{\sqrt{2\pi\sigma_{w,a}^2}} exp\{-\frac{1}{2}(\frac{(x_w-\mu_{w,a})^2}{\sigma_{w,a}^2})\}$$

$$= \frac{1}{2\pi * 50} exp\{-\frac{1}{2} * (50+50)\}$$

$$\approx 6.14 * 10^{-25}$$

$$p(x|y=c) \approx 5.56 * 10^{-7}$$

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

$$\approx 10^{-18}$$

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

$$\approx 1$$

- So the new data will be classified to child

Gaussian Naïve Bayes Classifier vs. LR

- For simplicity, we assume y is boolean, governed by a Bernoulli distribution, with parameter $\theta = P\ (y\ =\ 1)$.

Form of $P(y|\mathbf{x})$ for Gaussian Naïve Bayes Classifier

- For simplicity, we assume y is boolean, governed by a Bernoulli distribution, with parameter $\theta = P\ (y\ =\ 1)$.

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

$$= \frac{1}{1 + \frac{p(y = 0)p(\mathbf{x}|y = 0)}{p(y = 1)p(\mathbf{x}|y = 1)}} = \frac{1}{1 + \exp(\ln \frac{p(y = 0)p(\mathbf{x}|y = 0)}{p(y = 1)p(\mathbf{x}|y = 1)})}$$

$$= \frac{1}{1 + \exp(ln \frac{1 - \theta}{\theta} + \sum_{j=1}^{d} \ln \frac{p(xj|y = 0)}{p(xj|y = 1)})}$$

Form of $P(y|\mathbf{x})$ for Gaussian Naïve Bayes Classifier

- For simplicity, we assume y is boolean, governed by a Bernoulli distribution, with parameter $\theta = P\ (y\ =\ 1)$.

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

$$= \frac{1}{1+\frac{p(y=0)p(\mathbf{x}|y=0)}{p(y=1)p(\mathbf{x}|y=1)}} = \frac{1}{1+\exp(\ln\frac{p(y=0)p(\mathbf{x}|y=0)}{p(y=1)p(\mathbf{x}|y=1)})}$$

$$= \frac{1}{1+\exp(ln\frac{1-\theta}{\theta}+\sum_{j=1}^{d}\ln\frac{p(xj|y=0)}{p(xj|y=1)})}$$

Looks like w_0 in LR

Can we solve for w_i?

Form of $P(y|\mathbf{x})$ for Gaussian Naïve Bayes Classifier(cont.)

For each x_i, assume $P(x_i|Y = Ck)$ is a Gaussian distribution of the form $\mathcal{N}(\mu_{ik}, \sigma_i)$.

$$\ln \frac{p(x_j|y=0)}{p(x_j|y=1)} = \ln \frac{\frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{(x_j-\mu_{j0})^2}{2\sigma_j^2}\right)}{\frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{(x_j-\mu_{j1})^2}{2\sigma_j^2}\right)}$$

Form of $P(y|\mathbf{x})$ for Gaussian Naïve Bayes Classifier(cont.)

For each x_i, assume $P(x_i|Y = Ck)$ is a Gaussian distribution of the form $\mathcal{N}(\mu_{ik}, \sigma_i)$.

$$
\begin{aligned}
\ln \frac{p(x_j|y=0)}{p(x_j|y=1)} &= \ln \frac{\frac{1}{\sqrt{2\pi}\sigma_j} \exp(-\frac{(x_j-\mu_{j0})^2}{2\sigma_j^2})}{\frac{1}{\sqrt{2\pi}\sigma_j} \exp(-\frac{(x_j-\mu_{j1})^2}{2\sigma_i^2})} \\
&= \ln \exp(\frac{(x_j-\mu_{j1})^2 - (x_j-\mu_{j0})^2}{2\sigma_j^2}) \\
&= \frac{\mu_{j0}-\mu_{j1}}{\sigma_j^2} x_j + \frac{\mu_{j1}^2 - \mu_{j0}^2}{2\sigma_j^2}
\end{aligned}
$$

Form of $P(y|\mathbf{x})$ for Gaussian Naïve Bayes Classifier(cont.)

For each x_i, assume $P(x_i|Y = Ck)$ is a Gaussian distribution of the form $\mathcal{N}(\mu_{ik}, \sigma_i)$.

$$\ln \frac{p(x_j|y=0)}{p(x_j|y=1)} = \ln \frac{\frac{1}{\sqrt{2\pi}\sigma_j} \exp(-\frac{(x_j-\mu_{j0})^2}{2\sigma_j^2})}{\frac{1}{\sqrt{2\pi}\sigma_j} \exp(-\frac{(x_j-\mu_{j1})^2}{2\sigma_i^2})}$$

$$= \ln \exp(\frac{(x_j - \mu_{j1})^2 - (x_j - \mu_{j0})^2}{2\sigma_j^2})$$

$$= \frac{\mu_{j0} - \mu_{j1}}{\sigma_j^2} x_j + \frac{\mu_{j1}^2 - \mu_{j0}^2}{2\sigma_j^2}$$

Thus:

$$p(y=1|\mathbf{x}) = \frac{1}{1 + \exp(w_0 + \sum_{j=1}^{d} w_j x_j)}$$

Logistic Regression!

where, $w_0 = \ln \frac{1-\theta}{\theta} + \sum_{j=1}^{d} \frac{\mu_{j1}^2 - \mu_{j0}^2}{2\sigma_j^2}$, $\quad w_j = \frac{\mu_{j0} - \mu_{j1}}{\sigma_j^2}$

Gaussian Naïve Bayes vs. Logistic Regression

- Representation equivalence
  - But only in a special case! (GNB with class-independent variances)
- But what's the difference?
  - LR makes no assumptions about $P(\mathbf{x}|y)$ in learning!
  - Optimize different functions. Obtain different solutions.

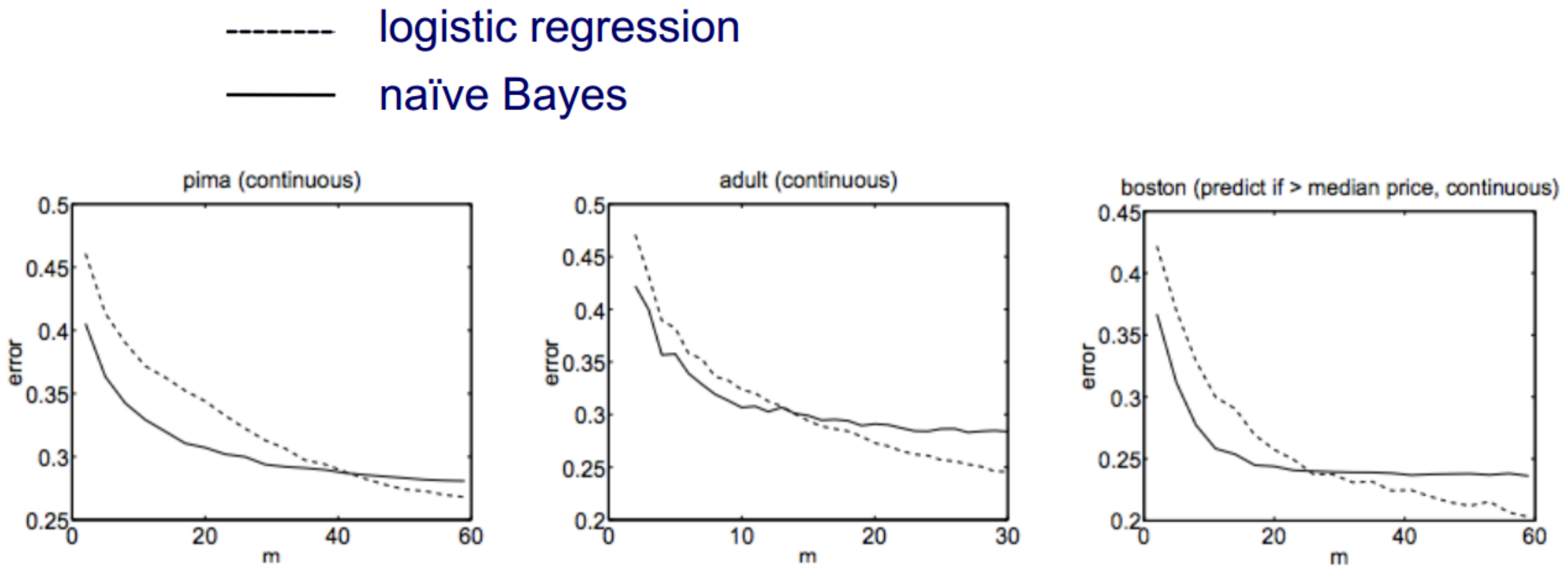
Gaussian Naïve Bayes vs. Logistic Regression

- Asymptotic comparison (#training examples $\rightarrow \infty$)
  - When model assumption is correct
    - GNB (with class independent variances) and LR produce identical classifiers.
  - When model assumption is not correct
    - LR is less biased, since it does not assume conditional independence.
    - LR expected to outperform NB when given lots of training data.

Gaussian Naïve Bayes vs. Logistic Regression

- Non-Asymptotic comparison (Ng and Jordan 2002)
  - Converge rate of parameter estimation: how many training example needed to assume good estimators?
    - NB: $O(\log d\ )$
    - LR: $O(d\ )$
    - d : dimension of sample **x**
  - NB converges much more quickly to its asymptotic estimators
  - NB expected to outperform LR with small training sets.

Experimental comparison of NB and LR



(Ng and Jordan) compared learning curves for the two approaches on 15 data sets.

General trend supports theory

- NB has lower predictive error when training sets are small.
- The error of LR approaches is lower than NB when training sets are large.

Generative vs. Discriminative Classifiers

Generative NB

- Assume functional form for
  - $p(\mathbf{x}|y)$ and $p(y)$
  - conditional independence
- Gaussian NB for continuous features
  - $p(x_j|y = C_k)$: $\mathcal{N}(\mu_{jk}, \sigma_k)$
  - $p(y)$: Bernoulli $(\theta, 1 - \theta)$
- Indirect computation
  - $p(y|\mathbf{x}) \propto p(\mathbf{x}|y)p(y)$

Discriminative LR

- Assume functional form for
  - $p(y|\mathbf{x})$
  - no assumptions
- Handles discrete & continuous features

$$\frac{1}{1 + \exp(w_0 + \sum_{j=1}^{d} w_j x_j)}$$

- Directly calculate $P(y|\mathbf{x})$

Discussion

- NB/LR is one case of a pair of generative/discriminative approaches for the same model class.
- If modeling assumptions are valid (*e.g.*, conditional independence of features in NB) the two will produce identical classifiers in the limit (# training instances $\rightarrow \infty$)
- If modeling assumptions are not valid, the discriminative approach is likely to be more accurate for large training sets.
- For small training sets, the generative approach is likely to be more accurate because parameters converge to their asymptotic values more quickly (in terms of training set size).

Tools for Naïve Bayes

scikit-learn

- a commonly used Python module for machine learning
- http://scikit-learn.org/stable/index.html
- Built on NumPy, SciPy, and matplotlib
- Simple and efficient tools for data mining and data analysis
- You can easily call the modules in sklearn to finish most machine learning tasks
- including logistic regression, Naïve Bayes, ridge regression, SVM, KNN,decision tree, random forest, GBDT …

NB in Sklearn

- an example of NB in sklearn
- 1、 import needed modules and prepare data

```
from sklearn import datasets, model_selection, naive_bayes
def load_data(datasets_name='iris'):
    if datasets_name == 'iris':
        data = datasets.load_iris()  # 加载 scikit-learn 自带的 iris 鸢尾花数据集-分类
    elif datasets_name == 'wine': # 0.18.2 没有
        data = datasets.load_wine()  # 加载 scikit-learn 自带的 wine 红酒起源数据集-分类
    elif datasets_name == 'cancer':
        data = datasets.load_breast_cancer()  # 加载 scikit-learn 自带的 乳腺癌数据集-分类
    else:
        pass

    return model_selection.train_test_split(data.data, data.target,test_size=0.25, random_state=0,stratify=data.target)
    # 分层采样拆分成训练集和测试集，测试集大小为原始数据集大小的 1/4
```

NB in Sklearn

- 2、 train the Gaussian Naïve Bayes model and test

```
def ttest_GaussianNB(X_train, X_test, y_train, y_test):
    cls = naive_bayes.GaussianNB()
    cls.fit(X_train, y_train)
    print('GaussianNB Testing Score: %.2f' % cls.score(X_test, y_test))
```

NB in Sklearn

- 3、 run the function and get results

```
for i in ['iris', 'wine', 'cancer']:
    print('\n====== %s ======\n' % i)
    X_train, X_test, y_train, y_test = load_data(datasets_name=i)  # 产生用于分类问题的数据集
    ttest_GaussianNB(X_train, X_test, y_train, y_test)
```

```
====== iris ======

GaussianNB Testing Score: 0.97

====== wine ======

GaussianNB Testing Score: 0.96

====== cancer ======

GaussianNB Testing Score: 0.92
```