Naïve Bayes Classifier

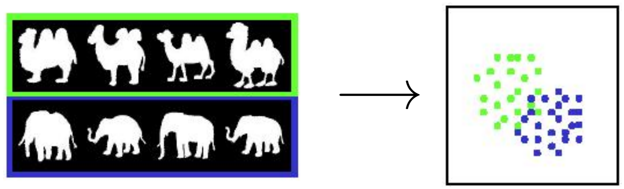
Lecture 7 (Part I), DD2431 Machine Learning

A. Maki

September 2014

Feature Space

- Sensors give *measurements* which can be converted to *features*.
- However in the real world



Samples **Feature space**

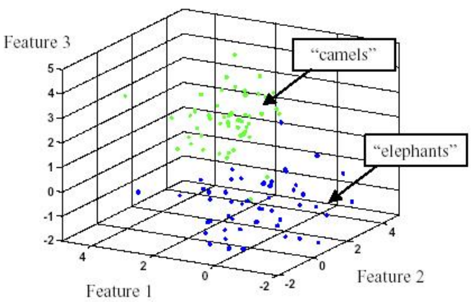
because of

- ✓ Measurement noise
- ✓ Intra-class variation
- ✓ Poor choice of features

Feature Space

End result: a $K-$dimensional space

- in which each dimension is a **feature**
- containing n labelled **samples** (objects)



Problem: Large Feature Space

- Size of feature space exponential in number of features.
- More features $\Longrightarrow$ potential for better description of the objects but...

  More features $\Longrightarrow$ more difficult to model $P(\mathbf{x} \mid y)$.
- **Extreme Solution:** Naïve Bayes classifier
  - ✓ All features (dimensions) regarded as independent.
  - ✓ Model k one-dimensional distributions instead of one k-dimensional distribution.

Naïve Bayes Classifier

- $\mathbf{x}$ is a vector $(x_1, \ldots, x_K)$ of attribute or feature values.
- Let $\mathcal{Y} = \{1, 2, \ldots, Y\}$ be the set of possible classes.
- The MAP estimate of y is

$$\begin{aligned} y_{\text{MAP}} &= \arg\max_{y \in \mathcal{Y}} P(y \mid x_1, \ldots, x_K) \\ &= \arg\max_{y \in \mathcal{Y}} \frac{P(x_1, \ldots, x_K \mid y)\, P(y)}{P(x_1, \ldots, x_K)} \\ &= \arg\max_{y \in \mathcal{Y}} P(x_1, \ldots, x_K \mid y)\, P(y) \end{aligned}$$

- Naïve Bayes assumption: $P(x_1, \ldots, x_K \mid y) = \prod_{k=1}^{K} P(x_k \mid y)$
- This give the *Naïve Bayes classifier*:

$$y_{\text{MAP}} = \arg\max_{y \in \mathcal{Y}} P(y) \prod_{k=1}^{K} P(x_k \mid y)$$

Naïve Bayes Classifier

- One of the most common learning methods.
- **When to use:**
  - ✓ Moderate or large training set available.
  - ✓ Features x_i of a data instance $\mathbf{x}$ are conditionally independent given classification (or at least reasonably independent, still works with a little dependence).
- **Successful applications:**
  - ✓ Medical diagnoses (symptoms independent)
  - ✓ Classification of text documents (words independent)

Example: Play Tennis?

Question: Will I go and play tennis given the forecast?

My measurements:

1. **forecast** $\in$ {sunny, overcast, rainy},
2. **temperature** $\in$ {hot, mild, cool},
3. **humidity** $\in$ {high, normal},
4. **windy** $\in$ {false, true}.

Possible decisions:

$y \in$ {yes, no}

Example: Play Tennis?

What I did in the past:

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes

outlook	temp.	humidity	windy	play
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Example: Play Tennis?

Counts of when I played tennis (did not play)

Outlook			Temperature			Humidity		Windy	
sunny	overcast	rain	hot	mild	cool	high	normal	false	true
2 (3)	4 (0)	3 (2)	2 (2)	4 (2)	3 (1)	3 (4)	6 (1)	6 (2)	3 (3)

Prior of whether I played tennis or not

Counts:

Play	
yes	no
9	5

Prior Probabilities:

Play	
yes	no
$\frac{9}{14}$	$\frac{5}{14}$

Likelihood of attribute when tennis played $P(x_i \mid \text{y=yes})(P(x_i \mid \text{y=no}))$

Outlook			Temperature			Humidity		Windy	
sunny	overcast	rain	hot	mild	cool	high	normal	false	true
$\frac{2}{9}$ $(\frac{3}{5})$	$\frac{4}{9}$ $(\frac{0}{5})$	$\frac{3}{9}$ $(\frac{2}{5})$	$\frac{2}{9}$ $(\frac{2}{5})$	$\frac{4}{9}$ $(\frac{2}{5})$	$\frac{3}{9}$ $(\frac{1}{5})$	$\frac{3}{9}$ $(\frac{4}{5})$	$\frac{6}{9}$ $(\frac{1}{5})$	$\frac{6}{9}$ $(\frac{2}{5})$	$\frac{3}{9}$ $(\frac{3}{5})$

Example: Play Tennis?

Inference: Use the learnt model to classify a new instance.

New instance:

$$\mathbf{x} = (\text{sunny, cool, high, true})$$

Apply Naïve Bayes Classifier:

$$y_{\text{MAP}} = \arg \max_{y \,\in\, \{\text{yes, no}\}} P(y) \prod_{i=1}^{4} P(x_i \mid y)$$

$$P(\text{yes})\, P(\text{sunny} \mid \text{yes})\, P(\text{cool} \mid \text{yes})\, P(\text{high} \mid \text{yes})\, P(\text{true} \mid \text{yes}) = \frac{9}{14} \times \frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} = .005$$

$$P(\text{no})\, P(\text{sunny} \mid \text{no})\, P(\text{cool} \mid \text{no})\, P(\text{high} \mid \text{no})\, P(\text{true} \mid \text{no}) = \frac{5}{14} \times \frac{3}{5} \times \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} = .021$$

$$\implies y_{\text{MAP}} = \text{no}$$

Naïve Bayes: Independence Violation

- Conditional independence assumption:

$$P(x_1, x_2, \ldots, x_K \mid y) = \prod_{k=1}^{K} P(x_k \mid y)$$

  often violated - but it works surprisingly well anyway!

- **Note:** Do not need the posterior probabilities $P(y \mid \mathbf{x})$ to be correct. Only need y_{MAP} to be correct.

- Since dependencies ignored, naïve Bayes posteriors often unrealistically close to 0 or 1.
  Different attributes say the same thing to a higher degree than we expect as they are correlated in reality.

Naïve Bayes: Estimating Probabilities

- **Problem:** What if none of the training instances with target value y have attribute x_i? Then

$$P(x_i \mid y) = 0 \quad \implies \quad P(y) \prod_{i=1}^{K} P(x_i \mid y) = 0$$

- **Solution:** Add as prior knowledge that $P(x_i \mid y)$ must be larger than 0:

$$P(x_i \mid y) = \frac{n_y + mp}{n + m}$$

where

n = number of training samples with label y
n_y = number of training samples with label y and value x_i
p = prior estimate of $P(x_i \mid y)$
m = weight given to prior estimate (in relation to data)

Example: Spam detection

- **Aim**: Build a classifier to identify spam e-mails.
- **How**:
  Training
  - ✓ Create dictionary of words and tokens $\mathcal{W} = \{w_1, \ldots, w_L\}$. These words should be those which are specific to spam or non-spam e-mails.
  - ✓ E-mail is a concatenation, in order, of its words and tokens: $\mathbf{e} = (e_1, e_2, \ldots, e_K)$ with $e_i \in \mathcal{W}$.
  - ✓ Must model and learn $P(e_1, e_2, \ldots, e_K \mid \text{spam})$ **and** $P(e_1, e_2, \ldots, e_K \mid \text{not spam})$

Email: E

Dear customer,
A fully licensed Online Pharmacy is offering pharmaceuticals:
- brought to you directly from abroad
-produced by the same multinational corporations selling through the major US pharmacies
-priced up to 5 times cheaper as compared to major US pharmacies.
Enjoy the US dollar purchasing power on http://pharmacy-buyonline.com.ua/

Vector: e

('dear', 'customer', ',', 'a', 'fully', 'licensed', ... ,'/')

Concatenate words from e-mail into a vector

Inference
  - ✓ Given an e-mail, E, compute $\mathbf{e} = (e_1, \ldots, e_K)$.
  - ✓ Use Bayes' rule to compute

$$P(\text{spam} \mid e_1, \ldots, e_K) \propto P(e_1, \ldots, e_K \mid \text{spam})\; P(\text{spam})$$

Example: Spam detection

- How is the joint probability distribution modelled?

$$P(e_1, \ldots, e_K \mid \text{spam})$$

  Remember K will be very large and vary from e-mail to e-mail..

- Make conditional independence assumption:

$$P(e_1, \ldots, e_K \mid \text{spam}) = \prod_{k=1}^{K} P(e_k \mid \text{spam})$$

  Similarly

$$P(e_1, \ldots, e_K \mid \text{not spam}) = \prod_{k=1}^{K} P(e_k \mid \text{not spam})$$

- Have assumed the position of word is not important.

Example: Spam detection

Learning:
Assume one has n training e-mails and their labels - spam /non-spam

$$\mathcal{S} = \{(\mathbf{e}_1, y_1), \ldots, (\mathbf{e}_n, y_n)\}$$

Note: $\mathbf{e}_i = (e_{i1}, \ldots, e_{iK_i})$.

Create dictionary

1. Make a union of all the distinctive words and tokens in $\mathbf{e}_1, \ldots, \mathbf{e}_n$ to create $\mathcal{W} = \{w_1, \ldots, w_L\}$. (**Note:** words such as *and*, *the*, ... omitted)

Learn probabilities

For $y \in \{\text{spam}, \text{not spam}\}$

1. Set $P(y) = \frac{\sum_{i=1}^n \text{Ind}(y_i = y)}{n}$ ← proportion of e-mails from class y.
2. $n_y = \sum_{i=1}^n K_i \times \text{Ind}\,(y_i = y)$ ← total # of words in the class y e-mails.
3. For each word w_l compute
   $n_{yl} = \sum_{i=1}^n \text{Ind}\,(y_i = y) \times \left(\sum_{k=1}^{K_i} \text{Ind}\,(e_{ik} = w_l)\right)$ ← # of occurrences of word w_l in the class y e-mails.
4. $P(w_l \mid y) = \frac{n_{yl}+1}{n_y + |\mathcal{W}|}$ ← assume prior value of $P(w_l \mid y)$ is $1/|\mathcal{W}|$.

Example: Spam detection

Inference: Classify a new e-mail $\mathbf{e}^* = (e_1^*, \ldots, e_{K^*}^*)$

$$y^* = \arg \max_{y \in \{-1,1\}} P(y) \prod_{k=1}^{K^*} P(e_k^* | y)$$