Machine learning

Presented by : Dr. Hanaa Bayomi

Lecture 7: Naïve bayse

Naïve bayes classifier

- It is a classification technique based on *Bayes theorem* with ***independent assumption among features (predictors)***.

- Naïve Bayes model is easy to build, with no complicated iterative parameter estimation ***which makes it particularly useful for very large datasets***

Bayes Theorem

- Given a class *C* and feature *X* which bears on the class:

$$P(C \mid X) = \frac{P(X \mid C)P(C)}{P(X)}$$

- *P(C)* : independent probability of *C (hypotheses)*: *prior probability*
- *P(X)* : independent probability of *X (data, predicator)*
- *P(X|C)*: conditional probability of *X* given C: *likelihood*
- *P(C|X)*: conditional probability of *C* given *X*: *posterior probability*

Maximum A Posterior

- Based on Bayes Theorem, we can compute the *Maximum A Posterior* (MAP) hypothesis for the data
- We are interested in the best hypothesis for some space C given observed training data X.

$$c_{MAP} \equiv \underset{c \in C}{\operatorname{argmax}} P(c \mid X)$$

$$= \underset{c \in C}{\operatorname{argmax}} \frac{P(X \mid c)P(c)}{P(X)}$$

$$= \underset{c \in C}{\operatorname{argmax}} P(X \mid c)P(c)$$

C: set of all hypothesis (Classes).

Note that we can drop *P(X)* as the probability of the data is constant (and independent of the hypothesis).

Bayes Classifiers

Assumption: training set consists of instances of different classes described *cj* as conjunctions of attributes values

Task: Classify a new instance *d* based on a tuple of attribute values into one of the classes $cj \in C$

Key idea: assign the most probable class c_{MAP} using Bayes Theorem.

$$c_{MAP} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j \mid x_1, x_2, \ldots, x_n)$$

$$= \underset{c_j \in C}{\operatorname{argmax}} \frac{P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)}{P(x_1, x_2, \ldots, x_n)}$$

$$= \underset{c_j \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)$$

The Naïve Bayes Model

- The *Naïve Bayes Assumption*: Assume that the effect of the value of the predictor (X) on a given class (C) is independent of the values of other predictors.

- This assumption is called class conditional independence

$$P(x_1, x_2, \ldots, x_n \mid C) = P(x_1 \mid C) \times P(x_2 \mid C) \times \ldots\ldots \times P(x_n \mid C)$$

$$P(x_1, x_2, \ldots, x_n \mid C) = \prod_{i=1}^{n} P(x_i \mid C)$$

Naïve Bayes Algorithm

- Naïve Bayes Algorithm (for discrete input attributes) has two phases

  - **1. Learning Phase**: Given a training set S,

    Learning is easy, just create probability tables.

    For each target value of c_i $(c_i = c_1, \cdots, c_L)$

    $\hat{P}(C = c_i) \leftarrow$ estimate $P(C = c_i)$ with examples in **S**;

    For every attribute value x_{jk} of each attribute X_j $(j = 1, \cdots, n; k = 1, \cdots, N_j)$

    $\hat{P}(X_j = x_{jk} \mid C = c_i) \leftarrow$ estimate $P(X_j = x_{jk} \mid C = c_i)$ with examples in **S**;

    Output: conditional probability tables; for $X_j, N_j \times L$ elements

  - **2. Test Phase**: Given an unknown instance $\mathbf{X}' = (a_1', \cdots, a_n')$,

    Look up tables to assign the label c^* to **X'** if

    $$[\hat{P}(a_1' \mid c^*) \cdots \hat{P}(a_n' \mid c^*)]\hat{P}(c^*) > [\hat{P}(a_1' \mid c) \cdots \hat{P}(a_n' \mid c)]\hat{P}(c), \quad c \neq c^*, c = c_1, \cdots, c_L$$

    Classification is easy, just multiply probabilities

Example

- Example: Play Tennis

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Example

- **Learning Phase**

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=Yes	Play=No
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Humidity	Play=*Yes*	Play=*No*
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

$P(\text{Play}=Yes) = 9/14$

$P(\text{Play}=No) = 5/14$

Example

- **Test Phase**

  – Given a new instance, predict its label

$\mathbf{x}'$=(Outlook=*Sunny*, Temperature=*Cool*, Humidity=*High*, Wind=*Strong*)

  – **Look up tables achieved in the learning phrase**

P(Outlook=*Sunny*|Play=*Yes*) = 2/9	P(Outlook=*Sunny*|Play=*No*) = 3/5
P(Temperature=*Cool*|Play=*Yes*) = 3/9	P(Temperature=*Cool*|Play==*No*) = 1/5
P(Huminity=*High*|Play=*Yes*) = 3/9	P(Huminity=*High*|Play=*No*) = 4/5
P(Wind=*Strong*|Play=*Yes*) = 3/9	P(Wind=*Strong*|Play=*No*) = 3/5
P(Play=*Yes*) = 9/14	P(Play=*No*) = 5/14

  – **Decision making with the MAP rule**

$P(Yes|\mathbf{x}') \approx [P(Sunny|Yes)P(Cool|Yes)P(High|Yes)P(Strong|Yes)]P(Play=Yes) = 0.0053$

$P(No|\mathbf{x}') \approx [P(Sunny|No)\ P(Cool|No)P(High|No)P(Strong|No)]P(Play=No) = 0.0206$

Given the fact $P(Yes|\mathbf{x}') < P(No|\mathbf{x}')$, we label $\mathbf{x}'$ to be "*No*".

Naïve Bayes

- Algorithm: Continuous-valued Features
  - Numberless values taken by a continuous-valued feature
  - Conditional probability often modeled with the normal distribution

$$\hat{P}(x_j \mid c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(x_j - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

μ_{ji} : mean (avearage) of feature values x_j of examples for which $c = c_i$

σ_{ji} : standard deviation of feature values x_j of examples for which $c = c_i$

  - Learning Phase: for $\mathbf{X} = (X_1, \cdots, X_n)$, $C = c_1, \cdots, c_L$

    Output: $n \times L$ normal distributions and $P(C = c_i)$ $i = 1, \cdots, L$

  - Test Phase: Given an unknown instance $\mathbf{X}' = (a'_1, \cdots, a'_n)$
    - Instead of looking-up tables, calculate conditional probabilities with all the normal distributions achieved in the learning phrase
    - Apply the MAP rule to assign a label (the same as done for the discrete case)

Naïve Bayes

- Example: Continuous-valued Features
  - Temperature is naturally of continuous value.

    Yes: 25.2, 19.3, 18.5, 21.7, 20.1, 24.3, 22.8, 23.1, 19.8

    No: 27.3, 30.1, 17.4, 29.5, 15.1
  - Estimate mean and variance for each class

$$\mu = \frac{1}{N}\sum_{n=1}^{N} x_n, \quad \sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2$$

$$\mu_{Yes} = 21.64, \ \sigma_{Yes} = 2.35$$
$$\mu_{No} = 23.88, \ \sigma_{No} = 7.09$$

  - **Learning Phase**: output two Gaussian models for P(temp|C)

$$\hat{P}(x \mid Yes) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x-21.64)^2}{2\times 2.35^2}\right) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x-21.64)^2}{11.09}\right)$$

$$\hat{P}(x \mid No) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x-23.88)^2}{2\times 7.09^2}\right) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x-23.88)^2}{50.25}\right)$$

Zero conditional probability

- If no example contains the feature value
  - In this circumstance, we face a zero conditional probability problem during test

$$\hat{P}(x_1 \mid c_i) \cdots \hat{P}(a_{jk} \mid c_i) \cdots \hat{P}(x_n \mid c_i) = 0 \qquad \text{for } x_j = a_{jk},\ \hat{P}(a_{jk} \mid c_i) = 0$$

  - For a remedy, class conditional probabilities re-estimated with

$$\hat{P}(a_{jk} \mid c_i) = \frac{n_c + mp}{n + m} \qquad \text{(m-estimate)}$$

n_c : number of training examples for which $x_j = a_{jk}$ and $c = c_i$

n : number of training examples for which $c = c_i$

p : prior estimate (usually, $p = 1/t$ for t possible values of x_j)

m : weight to prior (number of "virtual" examples, $m \geq 1$)

Zero conditional probability

- Example: P(outlook=overcast|no)=0 in the play-tennis dataset
  - Adding m "virtual" examples (m: up to 1% of #training example)
    - In this dataset, # of training examples for the "no" class is 5.
    - We can only add $m=1$ "virtual" example in our m-esitmate remedy.
  - The "outlook" feature can takes only 3 values. So $p=1/3$.
  - Re-estimate P(outlook|no) with the m-estimate

$$P(\text{overcast}|\text{no}) = \frac{0+1*\left(\frac{1}{3}\right)}{5+1} = \frac{1}{18}$$

$$P(\text{sunny}|\text{no}) = \frac{3+1*\left(\frac{1}{3}\right)}{5+1} = \frac{5}{9} \qquad P(\text{rain}|\text{no}) = \frac{2+1*\left(\frac{1}{3}\right)}{5+1} = \frac{7}{18}$$

$$\hat{P}(a_{jk} \mid c_i) = \frac{n_c + mp}{n + m}$$

n_c : 0 (No.of samples outlook=overcast|no)

n : 5 (No.of samples class=no)

p : 1/3 (outloock has 3 values(sunny, overcast, rain))

m : 1

Conclusion

- **Naïve Bayes is based on the** independence assumption
- Training is very easy and fast; just requiring considering each attribute in each class separately
- Test is straightforward; just looking up tables or calculating conditional probabilities with normal distributions

- **Naïve Bayes**

- Performance of naïve Bayes is competitive to most of state-of-the-art classifiers even if in presence of violating the independence assumption
- It has many successful applications, e.g., spam mail filtering