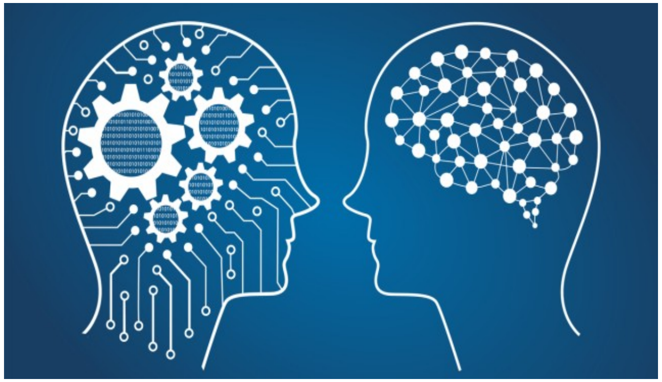

Naïve Bayes Classification

Dr. Ammar Mohammed
Associate Professor of Computer Science
ISSR, Cairo University
PhD of CS (Uni. Koblenz-Landau, Germany)
Spring 2019

Contact:
mailto: Ammar@cu.edu.eg
Drammarcu@gmail.com

Naive Bayes Classifier

Can be used successfully in varieties of applications

- **Text Classification**

  Whether a text document belongs to one or more categories (classes)

- **Spam Filtering**

  Given an email, predicts whether it is spam or not

- **Sentiment Analysis**

  Analyze the tones of tweets, comments and reviews, and predict whether they are negative, positive or neutral

- **Recommendation Systems**

  With combination with collaborative filtering, naive classifier is used to build hybrid system for recommendation products

- **Medical Diagnosis**

  Given a list of symptoms, predict whether a patient has disease X or not

Probability Basics

Probability of an Event E, denoted as P(E),

$$P(E)=\frac{n(E)}{n}$$

Number of occurrences of E

Sample space
(total number of possible outcomes)

Conditional Probability

$$P(A|B)=\frac{n(A,B)}{n(B)}$$

Joint Probability

$$P(A,B)=P(A|B)\,P(B)$$

The product Rule

Bayes Rule

Starting with the product rule

$$P(A,B) = P(A|B)\,P(B)$$ **(1)**

We can swap B and A

$$P(B,A) = P(B|A)\,P(A)$$ **(2)**

The symmetry rule tells us that P (A, B) = P (B, A). by (1) and (2)

$$P(B|A)\,P(A) = P(A|B)\,P(B)$$

$$P(B|A) = \frac{P(A|B)\,P(B)}{P(A)}$$ ← This is Bayes Theorem

$$Posterior = \frac{Likelihood \times Prior}{Evidence}$$

Computing the Posterior

Let H be a *hypothesis* that X belongs to class $C_i \in C = \{C_1, C_2, \ldots, C_k\}$

- Given training data **X**, *posteriori probability of a hypothesis* H, P(H|**X**), follows the Bayes theorem

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

Predicts **X** belongs to C_i **iff** the probability P(H=C_i|**X**) is the highest among all the P(H=C_k|X) for all the *k classes*

- Practical difficulty: require initial knowledge of many probabilities, significant computational cost

Maximum Posteriori Estimation MPE

Our prediction is the value of C_i, which maximizes the posterior distribution.

$$C_{MPE} = \arg\max_{c_i \in C} \frac{P(X \mid H = C_i)\, P(H = C_i)}{P(X)}$$

But P (x) is always positive and doesn't depend on C. If we are only looking at what maximizes the posterior, we can safely discard it. Thus, we can make a prediction for the class using only

$$C_{MPE} = \arg\max_{c_i \in C} P(X \mid H = C_i)\, P(H = C_i)$$

Why is Naïve Bayes "naïve"

$$C_{MPE} = arg\ max_{c_i \in C} P(X \mid H = C_i) P(H = C_i)$$

What if X =(x_1,x_2,…,x_d) is a vector of features with dimension d? We'll then have to compute **P (X|H=c_i) P (H=c_i)** for each $c_i \in C$

Difficulty: learning the joint probability is infeasible

The problem with explicitly modeling P(X_1,…,X_d|H=c_i) is that there are too many parameters:

$$P(x_1, x_{2,\dots}, x_d \mid H = c_i) = P(x_1 \mid x_{2,\dots}, x_d, H = c_i) P(x_2 \mid x_{3,\dots}, x_d, H = c_i) \dots P(x_d \mid H = c_i) P(H = c_i)$$

run out of space, run out of time, and need tons of training data (which is usually not available)

Naïve Bayes Model

- ***The Naïve Bayes Assumption***: Assume that all features are **independent** given the class label C
- Equationally speaking:

$$P(X|H) = P(x_1|H)\,P(x_2|H)\ldots P(x_d|H)$$

$$P(X|H) = \prod_{i=1}^{d} P(x_i|H)$$

If x_k categorical, $P(x_k|C_i)$ is the number of tuples in C_i having value x_k , divided by $|C_{i, D}|$ (number of tuples of C_i in the data set D)

Naïve Bayes Algorithm

- **Algorithm On Discrete valued Features**

Learning Phase: Given a training set **M** of ***d*** features and $Y=\{c_1,c_2,\dots,c_k\}$ classes

> For each target value class $c_j \in Y$
> $\hat{P}(c_j) \longleftarrow$ estimate $P(c_j)$ with examples in **M**
>   For every feature value X_{ik} of each feature X_i (j=1,..m; k=1,.d)
> $\hat{P}(X_i=x_{ik} \mid c_j) \leftarrow$ estimate $P(x_{ik} \mid c_j)$ with examples in M

Output: Conditional Probabilistic (generative) model

Prediction Phase: Given a new input feature $\mathbf{X}=(a_1,a_2,\dots,a_d)$

Lookup tables: to assign the class $\mathbf{c}^*$ to **X** having

$$[\hat{P}(a_1 \mid c^*)\ \hat{P}(a_2 \mid c^*) .. \hat{P}(a_d \mid c^*)]\ \hat{P}(c^*) > [\hat{P}(a_1 \mid c_i)\ \hat{P}(a_2 \mid c_i) .. \hat{P}(a_d \mid c_i)]\ \hat{P}(c_i)$$

for all $c_i=c_1,c_2, \dots, c_k$

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

- Prediction Phase
  - Given a new instance, predict its label

    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(Temperature=*Cool*|Play=*Yes*) = 3/9
P(Humidity=*High*|Play=*Yes*) = 3/9
P(Wind=*Strong*|Play=*Yes*) = 3/9
P(Play=*Yes*) = 9/14

P(Outlook=*Sunny*|Play=*No*) = 3/5
P(Temperature=*Cool*|Play==*No*) = 1/5
P(Humidity=*High*|Play=*No*) = 4/5
P(Wind=*Strong*|Play=*No*) = 3/5
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 **x** to be "*No*".

Naive Bayes with Continuous Features

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

$$P(x_i|c_j) = \frac{1}{\sigma_{ji}\sqrt{2\pi}} e^{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ji}^2}}$$

μ_{ij} :mean of feature values x_i of examples for which $c=c_j$

σ_{ji}^2 :standard deviation of feature values x_i of examples for which $c=c_j$

Prediction Phase: Given a new input feature **X**=(a_1,a_2,…,a_d)

- Instead of looking-up tables, calculate conditional probabilities with all the normal distributions achieved in the learning phrase
- Apply the rule to assign a label (the same as done for the discrete case)

Naive Bayes with Continuous Features

- Example: Continuous-valued Features
  - Temperature Feature 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

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \sigma^2 = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \mu\right)^2$$

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

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

$$P\left(x \mid Yes\right) = \frac{1}{2.35\sqrt{2\pi}} e^{-\frac{\left(x-21.64\right)^2}{11.09}} \qquad P\left(x \mid No\right) = \frac{1}{7.09\sqrt{2\pi}} e^{-\frac{\left(x-23.88\right)^2}{50.25}}$$

Zero Conditional Probability Problem

- Naïve Bayesian prediction requires each conditional prob. be non-zero. Otherwise, the predicted prob. will be zero

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$

- Ex. Suppose a dataset with 1000 tuples, income=low (0), income= medium (990), and income = high (10),
- **Use Laplacian correction (or Laplacian estimator)**
  - Adding 1 to each case

    Prob(income = low) = 1/1003

    Prob(income = medium) = 991/1003

    Prob(income = high) = 11/1003
  - The "corrected" prob. estimates are close to their "uncorrected" counterparts

Naïve Bayesian Classifier: Comments

- Advantages
  - Easy to implement
  - Good results obtained in most of the cases
- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., hospitals: patients: Profile: age, family history, etc.

      Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayesian Classifier