Machine Learning

(IT3190E)

Quang Nhat NGUYEN

quang.nguyennhat@hust.edu.vn

Hanoi University of Science and Technology

School of Information and Communication Technology

Academic year 2020-2021

The course's content:

- Introduction
- Performance evaluation of ML system
- **Supervised learning**
  - **Probabilistic learning**
- Unsupervised learning
- Ensemble learning
- Reinforcement learning

Probabilistic learning

- Statistical approaches for the learning problem
- In this lecture, we focus on the classification problem
  - Classification is done based on a statistical model
  - Classification is done based on the probabilities of the possible class labels
- Main topics:
  - Introduction of probability theorem
  - Bayes theorem
  - Maximum a posteriori
  - Maximum likelihood estimation
  - Naïve Bayes classification

Basic probability concepts

- Suppose we have an experiment (e.g., a dice roll) whose outcome depends on chance
- *Sample space* `S`. A set of all possible outcomes

  E.g., `S= {1,2,3,4,5,6}` for a dice roll
- *Event* `E`. A subset of the sample space

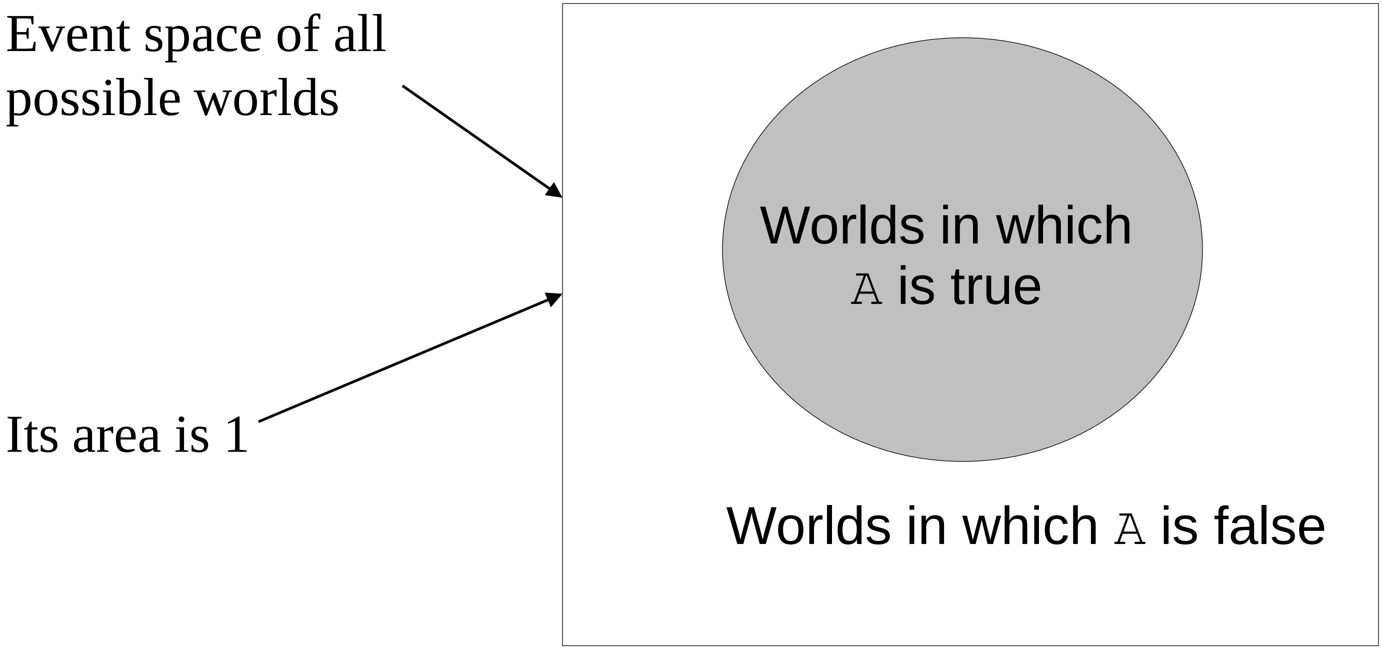
  E.g., `E= {1}:` the result of the roll is one

  E.g., `E= {1,3,5}:` the result of the roll is an odd number
- *Event space* `W`. The possible worlds the outcome can occur

  E.g., `W` includes all dice rolls
- *Random variable* `A`. A random variable represents an event, and there is some degree of chance (probability) that the event occurs

Visualizing probability

`P(A)`: "the fraction of possible worlds in which `A` is true"


Event space of all possible worlds
Its area is 1
Worlds in which A is true
Worlds in which A is false

[*http://www.cs.cmu.edu/~awm/tutorials*]

Boolean random variables

- A Boolean random variable can take either of the two Boolean values, `true` or `false`
- The axioms
  - $0 \leq P(A) \leq 1$
  - $P(true) = 1$
  - $P(false) = 0$
  - $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$
- The corollaries
  - $P(\text{not } A) \equiv P(\sim A) = 1 - P(A)$
  - $P(A) = P(A \wedge B) + P(A \wedge \sim B)$

Multi-valued random variables

A multi-valued random variable can take a value from a set of `k` (>2) values $\{v_1, v_2, \ldots, v_k\}$

$$P(A = v_i \wedge A = v_j) = 0 \text{ if } i \neq j$$

$$P(A=v_1 \vee A=v_2 \vee \ldots \vee A=v_k) = 1$$

$$P(A = v_1 \vee A = v_2 \vee \ldots \vee A = v_i) = \sum_{j=1}^{i} P(A = v_j)$$

$$\sum_{j=1}^{k} P(A = v_j) = 1$$

$$P\left(B \wedge \left[A = v_1 \vee A = v_2 \vee \ldots \vee A = v_i\right]\right) = \sum_{j=1}^{i} P(B \wedge A = v_j)$$

[*http://www.cs.cmu.edu/~awm/tutorials*]

Conditional probability (1)

- `P(A|B)` is the fraction of worlds in which `A` is true given that `B` is true
- Example
  - `A`: I will go to the football match tomorrow
  - `B`: It will be not raining tomorrow
  - `P(A|B)`: The probability that I will go to the football match if (given that) it will be not raining tomorrow

Conditional probability (2)

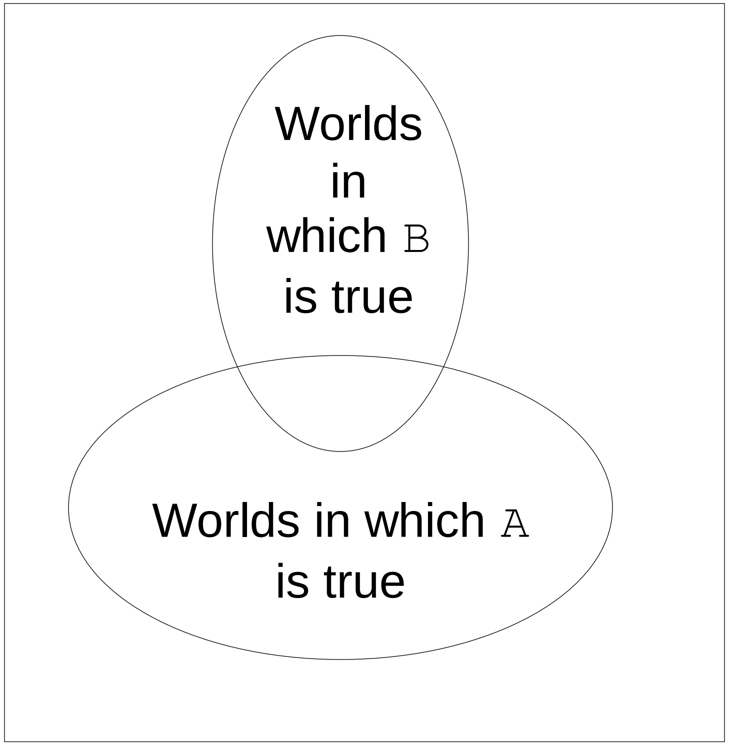
Definition: $$P(A|B) = \frac{P(A,B)}{P(B)}$$

Corollaries:

`P(A,B)=P(A|B).P(B)`

`P(A|B)+P(~A|B)=1`

$$\sum_{i=1}^{k} P(A=v_i \mid B) = 1$$



Independent variables (1)

- Two events A and B are ***statistically independent*** if the probability of A is the same value
  - when B occurs, or
  - when B does not occur, or
  - when nothing is known about the occurrence of B

- Example
  - A: I will play a football match tomorrow
  - B: Bob will play the football match
  - P(A|B) = P(A)

    → "Whether Bob will play the football match tomorrow does not influence my decision of going to the football match."

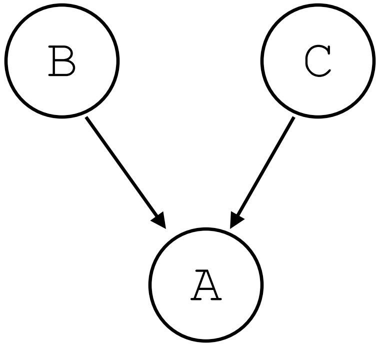
Independent variables (2)

From the definition of independent variables P(A|B)=P(A), we can derive the following rules

- P(~A|B) = P(~A)
- P(B|A) = P(B)
- P(A,B) = P(A). P(B)
- P(~A,B) = P(~A). P(B)
- P(A,~B) = P(A). P(~B)
- P(~A,~B) = P(~A). P(~B)

Conditional probability for >2 variables

- P(A|B,C) is the probability of A given B and C
- Example
  - A: I will walk along the river tomorrow morning
  - B: The weather is beautiful tomorrow morning
  - C: I will get up early tomorrow morning
  - P(A|B,C): The probability that I will walk along the river tomorrow morning if (given that) the weather is nice and I get up early



P(A|B,C)

Conditional independence

- Two variables `A` and `C` are ***conditionally independent*** given variable `B` if the probability of `A` given `B` is the same as the probability of `A` given `B` and `C`
- Formal definition: `P(A|B,C) = P(A|B)`
- Example
  - `A`: I will play the football match tomorrow
  - `B`: The football match will take place indoor
  - `C`: It will be not raining tomorrow
  - `P(A|B,C)=P(A|B)`

    → Given knowing that the match will take place indoor, the probability that I will play the match does not depend on the weather

Probability – Important rules

- Chain rule
  - $P(A,B) = P(A|B).P(B) = P(B|A).P(A)$
  - $P(A|B) = P(A,B)/P(B) = P(B|A).P(A)/P(B)$
  - $P(A,B|C) = P(A,B,C)/P(C) = P(A|B,C).P(B,C)/P(C)$

    $= P(A|B,C).P(B|C)$
- (Conditional) independence
  - $P(A|B) = P(A)$; if A and B are independent
  - $P(A,B|C) = P(A|C).P(B|C)$; if A and B are conditionally independent given C
  - $P(A_1,\ldots,A_n|C) = P(A_1|C)\ldots P(A_n|C)$; if $A_1,\ldots,A_n$ are conditionally independent given C

Bayes theorem

$$P(h \mid D) = \frac{P(D \mid h).P(h)}{P(D)}$$

- `P(h)`: Prior probability of hypothesis (e.g., classification) `h`
- `P(D)`: Prior probability that the data `D` is observed
- `P(D|h)`: Probability of observing the data `D` given hypothesis `h`
- `P(h|D)`: Probability of hypothesis `h` given the observed data `D`
  - ➢ **Probabilistic classification methods use this this *posterior probability*!**

Bayes theorem – Example (1)

Assume that we have the following data (of a person):

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
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

[Mitchell, 1997]

Bayes theorem – Example (2)

- Dataset `D`. The data of the days when the outlook is sunny and the wind is strong
- Hypothesis `h`. The person plays tennis
- Prior probability `P(h)`. Probability that the person plays tennis (i.e., irrespective of the outlook and the wind)
- Prior probability `P(D)`. Probability that the outlook is sunny and the wind is strong
- `P(D|h)`. Probability that the outlook is sunny and the wind is strong, given knowing that the person plays tennis
- `P(h|D)`. Probability that the person plays tennis, given knowing that the outlook is sunny and the wind is strong

  → We are interested in this *posterior probability*!!

Maximum a posteriori (MAP)

- Given a set H of possible hypotheses (e.g., possible classifications), the learner finds ***the most probable hypothesis*** h(∈H) given the observed data D

- Such a maximally probable hypothesis is called a maximum a posteriori (MAP) hypothesis

$$h_{MAP} = \arg\max_{h \in H} P(h \mid D)$$

$$h_{MAP} = \arg\max_{h \in H} \frac{P(D \mid h).P(h)}{P(D)}$$

(by Bayes theorem)

$$h_{MAP} = \arg\max_{h \in H} P(D \mid h).P(h)$$

(P(D) is a constant, independent of h)

MAP hypothesis – Example

- The set H contains two hypotheses
  - h_1: The person will play tennis
  - h_2: The person will not play tennis
- Compute the two posteriori probabilities $P(h_1|D)$, $P(h_2|D)$
- The MAP hypothesis: $h_{MAP}=h_1$ if $P(h_1|D) \geq P(h_2|D)$; otherwise $h_{MAP}=h_2$
- Because $P(D)=P(D,h_1)+P(D,h_2)$ is the same for both h_1 and h_2, we ignore it
- So, we compute the two formulae: $P(D|h_1).P(h_1)$ and $P(D|h_2).P(h_2)$, and make the conclusion:
  - If $P(D|h_1).P(h_1) \geq P(D|h_2).P(h_2)$, the person will play tennis;
  - Otherwise, the person will not play tennis

Maximum likelihood estimation (MLE)

- Phương pháp MAP: Với một tập các giả thiết có thể H, cần tìm một giả thiết cực đại hóa giá trị: P(D|h).P(h)
- Giả sử (assumption) trong phương pháp **đánh giá khả năng có thể nhất (Maximum likelihood estimation – MLE)**: Tất cả các giả thiết đều có giá trị xác suất trước như nhau: $P(h_i)=P(h_j)$, $\forall h_i,h_j \in H$
- Phương pháp MLE tìm giả thiết cực đại hóa giá trị P(D|h); trong đó P(D|h) được gọi là *khả năng có thể (likelihood)* của dữ liệu D đối với h
- Giả thiết có khả năng nhất (maximum likelihood hypothesis)

$$h_{ML} = \arg\max_{h \in H} P(D \mid h)$$

ML hypothesis – Example

- The set H contains two hypotheses
  - h_1: The person will play tennis
  - h_2: The person will not play tennis

  D: The data of the dates when the outlook is sunny and the wind is strong

- Compute the two likelihood values of the data D given the two hypotheses: $P(D|h_1)$ and $P(D|h_2)$
  - $P(Outlook=Sunny, Wind=Strong|h_1) = 1/8$
  - $P(Outlook=Sunny, Wind=Strong|h_2) = 1/4$

- The ML hypothesis $h_{ML}=h_1$ if $P(D|h_1) \geq P(D|h_2)$; otherwise $h_{ML}=h_2$

  → Because $P(Outlook=Sunny, Wind=Strong|h_1) < P(Outlook=Sunny, Wind=Strong|h_2)$, we arrive at the conclusion: *The person will not play tennis*

Naïve Bayes classifier (1)

- Problem definition
  - A training set `D`, where each training instance `x` is represented as an `n`-dimensional attribute vector: $(x_1, x_2, ..., x_n)$
  - A pre-defined set of classes: $C=\{c_1, c_2, ..., c_m\}$
  - Given a new instance `z`, which class should `z` be classified to?
- We want to find the most probable class for instance `z`

$$c_{MAP} = \arg\max_{c_i \in C} P(c_i \mid z)$$

$$c_{MAP} = \arg\max_{c_i \in C} P(c_i \mid z_1, z_2, ..., z_n)$$

$$c_{MAP} = \arg\max_{c_i \in C} \frac{P(z_1, z_2, ..., z_n \mid c_i).P(c_i)}{P(z_1, z_2, ..., z_n)}$$ (by Bayes theorem)

Naïve Bayes classifier (2)

- To find the most probable class for z (continued…)

$$c_{MAP} = \arg\max_{c_i \in C} P(z_1, z_2, \ldots, z_n \mid c_i).P(c_i)$$

(P(z1,z2,...,zn) is the same for all classes)

- **Assumption in Naïve Bayes classifier**. The attributes are *conditionally independent* given classification

$$P(z_1, z_2, \ldots, z_n \mid c_i) = \prod_{j=1}^{n} P(z_j \mid c_i)$$

- Naïve Bayes classifier finds the most probable class for z

$$c_{NB} = \arg\max_{c_i \in C} P(c_i).\prod_{j=1}^{n} P(z_j \mid c_i)$$

Naïve Bayes classifier - Algorithm

- The learning (training) phase (given a training set)

  For each classification (i.e., class label) $c_i \in C$

  - Estimate the priori probability: $P(c_i)$
  - For each attribute value x_j, estimate the probability of that attribute value given classification c_i: $P(x_j | c_i)$

- The classification phase (given a new instance)
  - For each classification $c_i \in C$, compute the formula

$$P(c_i).\prod_{j=1}^{n} P(x_j | c_i)$$

  - Select the most probable classification c^*

$$c^* = \arg\max_{c_i \in C} P(c_i).\prod_{j=1}^{n} P(x_j | c_i)$$

Naïve Bayes classifier – Example (1)

Will a young student with medium income and fair credit rating buy a computer?

Rec. ID	Age	Income	Student	Credit_Rating	Buy_Computer
1	Young	High	No	Fair	No
2	Young	High	No	Excellent	No
3	Medium	High	No	Fair	Yes
4	Old	Medium	No	Fair	Yes
5	Old	Low	Yes	Fair	Yes
6	Old	Low	Yes	Excellent	No
7	Medium	Low	Yes	Excellent	Yes
8	Young	Medium	No	Fair	No
9	Young	Low	Yes	Fair	Yes
10	Old	Medium	Yes	Fair	Yes
11	Young	Medium	Yes	Excellent	Yes
12	Medium	Medium	No	Excellent	Yes
13	Medium	High	Yes	Fair	Yes
14	Old	Medium	No	Excellent	No

http://www.cs.sunysb.edu/~cse634/lecture_notes/07classification.pdf

Naïve Bayes classifier – Example (2)

- Representation of the problem
  - x = (Age=Young,Income=Medium,Student=Yes,Credit_Rating=Fair)
  - Two classes: c_1 (buy a computer) and c_2 (not buy a computer)
- Compute the priori probability for each class
  - P(c_1) = 9/14
  - P(c_2) = 5/14
- Compute the probability of each attribute value given each class

P(Age=Young|c_1) = 2/9;	P(Age=Young|c_2) = 3/5
P(Income=Medium|c_1) = 4/9;	P(Income=Medium|c_2) = 2/5
P(Student=Yes|c_1) = 6/9;	P(Student=Yes|c_2) = 1/5
P(Credit_Rating=Fair|c_1) = 6/9;	P(Credit_Rating=Fair|c_2) = 2/5

Naïve Bayes classifier – Example (3)

- Compute the likelihood of instance $\mathtt{x}$ given each class
  - For class $\mathtt{c_1}$

    $P(\mathtt{x}|\mathtt{c_1}) = P(\text{Age}=\mathtt{Young}|\mathtt{c_1}).P(\text{Income}=\mathtt{Medium}|\mathtt{c_1}).P(\text{Student}=\mathtt{Yes}|\mathtt{c_1}).P(\text{Credit\_Rating}=\mathtt{Fair}|\mathtt{c_1}) = (2/9).(4/9).(6/9).(6/9) = 0.044$

  - For class $\mathtt{c_2}$

    $P(\mathtt{x}|\mathtt{c_2}) = P(\text{Age}=\mathtt{Young}|\mathtt{c_2}).P(\text{Income}=\mathtt{Medium}|\mathtt{c_2}).P(\text{Student}=\mathtt{Yes}|\mathtt{c_2}).P(\text{Credit\_Rating}=\mathtt{Fair}|\mathtt{c_2}) = (3/5).(2/5).(1/5).(2/5) = 0.019$

- Find the most probable class
  - For class $\mathtt{c_1}$

    $P(\mathtt{c_1}).P(\mathtt{x}|\mathtt{c_1}) = (9/14).(0.044) = 0.028$

  - For class $\mathtt{c_2}$

    $P(\mathtt{c_2}).P(\mathtt{x}|\mathtt{c_2}) = (5/14).(0.019) = 0.007$

  → Conclusion: *The person* $\mathtt{x}$ *will buy a computer*!

Naïve Bayes classifier – Issues (1)

- What happens if no training instances associated with class c_i have attribute value x_j?

  $P(x_j|c_i)=0$, and hence: $P(c_i).\prod_{j=1}^{n} P(x_j \mid c_i) = 0$

- Solution: to use a Bayesian approach to estimate $P(x_j|c_i)$

  $$P(x_j \mid c_i) = \frac{n(c_i, x_j) + mp}{n(c_i) + m}$$

  - $n(c_i)$: number of training instances associated with class c_i
  - $n(c_i, x_j)$: number of training instances associated with class c_i that have attribute value x_j
  - p: a prior estimate for $P(x_j|c_i)$

    → Assume uniform priors: $p=1/k$, if attribute f_j has k possible values
  - m: a weight given to prior

    → To augment the $n(c_i)$ actual observations by an additional m virtual samples distributed according to p

Naïve Bayes classifier – Issues (2)

- The limit of precision in computers' computing capability
  - $P(x_j|c_i)$<1, for every attribute value x_j and class c_i
  - So, when the number of attribute values is very large

$$\lim_{n\to\infty}\left(\prod_{j=1}^{n} P(x_j\,|\,c_i)\right) = 0$$

- Solution: to use a logarithmic function of probability

$$c_{NB} = \arg\max_{c_i \in C}\left(\log\left[P(c_i).\prod_{j=1}^{n} P(x_j\,|\,c_i)\right]\right)$$

$$c_{NB} = \arg\max_{c_i \in C}\left(\log P(c_i) + \sum_{j=1}^{n} \log P(x_j\,|\,c_i)\right)$$

Document classification by NB – Training

- Problem definition
  - A training set D, where each training example is a document associated with a class label: D = {(d_k, c_i)}
  - A pre-defined set of class labels: C = {c_i}
- The training algorithm
  - From the documents collection contained in the training set D, extract the vocabulary of distinct terms (keywords): T = {t_j}
  - Let's denote D_c_i (⊆D) the set of documents in D whose class label is c_i
  - For each class c_i
    - Compute the priori probability of class c_i: $P(c_i)=\frac{|D\_c_i|}{|D|}$
    - For each term t_j, compute the probability of term t_j given class c_i

$$P(t_j \mid c_i)=\frac{\left(\sum_{d_k \in D\_c_i} n(d_k,t_j)\right)+1}{\left(\sum_{d_k \in D\_c_i} \sum_{t_m \in T} n(d_k,t_m)\right)+|T|}$$

($n(d_k, t_j)$: the number of occurrences of term t_j in document d_k)

Document classification by NB – Classifying

- To classify (assign the class label for) a new document d
- The classification algorithm
  - From the document d, extract a set T_d of all terms (keywords) t_j that are known by the vocabulary T (i.e., T_d $\subseteq$ T)
  - **Additional assumption**. The probability of term t_j given class c_i is independent of its position in document

    P(t_j at position k | c_i) = P(t_j at position m | c_i), $\forall$ k,m
  - For each class c_i, compute the likelihood of document d given c_i

$$P(c_i).\prod_{t_j \in T\_d} P(t_j \mid c_i)$$

  - Classify document d in class c^*

$$c^* = \arg\max_{c_i \in C} P(c_i).\prod_{t_j \in T\_d} P(t_j \mid c_i)$$

Naïve Bayes classifier – Summary

- One of the most practical learning methods
- Based on the Bayes theorem
- Very fast in performance
  - For the training: only one pass over (scan through) the training set
  - For the classification: the computation time is linear in the number of attributes and the size of the documents collection
- Despite its conditional independence assumption, Naïve Bayes classifier shows a good performance in several application domains
- When to use?
  - A moderate or large training set available
  - Instances are represented by a large number of attributes
  - **Attributes** that describe instances **are conditionally independent given classification**