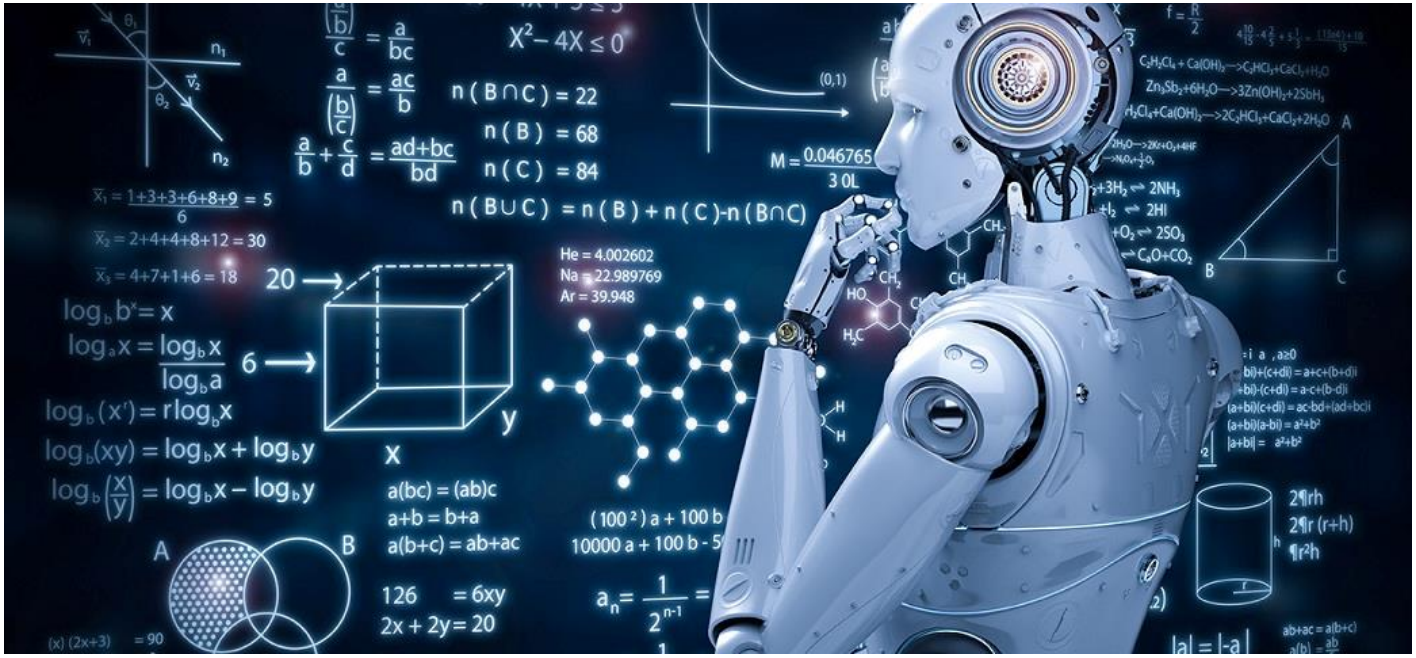


BBM406: Fundamentals of Machine Learning

Bayesian Learning

Bayesian Learning

Features of Bayesian learning methods:

- Bayesian methods can accommodate hypotheses that make probabilistic predictions.
- New instances can be classified by combining the predictions of multiple hypotheses, weighted by their probabilities.
- Even in cases where Bayesian methods prove computationally intractable, they can provide a standard of optimal decision making.

Bayesian Learning

Features of Bayesian learning methods:

- Each observed training example can incrementally decrease or increase the estimated probability of the correctness of a hypothesis.
  - This provides a more flexible approach to learning than algorithms that completely eliminate a hypothesis if it is found to be inconsistent with any single example.
- Prior knowledge can be combined with observed data to determine the final probability of a hypothesis.
- In Bayesian learning, prior knowledge is provided by asserting
  - a prior probability for each candidate hypothesis, and
  - a probability distribution over observed data for each possible hypothesis.

Difficulties with Bayesian Methods

- Require initial knowledge of many probabilities
  - When these probabilities are not known in advance they are often estimated based on background knowledge, previously available data, and assumptions about the form of the underlying distributions.
- Significant computational cost is required to determine the Bayes optimal hypothesis in the general case (linear in the number of candidate hypotheses).
  - In certain specialized situations, this computational cost can be significantly reduced.

Bayes Theorem

- P(A) is **prior probability** (**unconditional probability**) of event A.

- P(A|B) is **posterior probability** (**conditional probability**) of event A given that event B holds.

- P(A,B) is the **joint probability** of two events A and B.
  - The (unconditional) probability of the events A and B occurring together.
  - P(A,B) = P(B,A)

Bayes Theorem

$P(A|B) = P(A,B) / P(B)$ ➔ $P(A,B) = P(A|B)*P(B)$

$P(B|A) = P(B,A) / P(A)$ ➔ $P(B,A) = P(B|A)*P(A)$

Since $P(A,B) = P(B,A)$, we have $P(A|B)*P(B) = P(B|A)*P(A)$

Thus, we have **Bayes Theorem**

$P(A|B) = P(B|A)*P(A) / P(B)$

$P(B|A) = P(A|B)*P(B) / P(A)$

Bayes Theorem - Example

Bayes Theorem

P(A|B) = P(B|A)*P(A) / P(B)

P(B|A) = P(A|B)*P(B) / P(A

Sample Space for events A and B

A holds	T	T	F	F	T	F	T
B holds	T	F	T	F	T	F	F

P(A) = 4/7 P(B) = 3/7 P(A,B) = P(B,A) = 2/7

P(B|A) = 2/4 P(A|B) = 2/3

Is Bayes Theorem correct?

P(B|A) = P(A|B)*P(B) / P(A) = (2/3 * 3/7) / 4/7 = 2/4 ➔ CORRECT

P(A|B) = P(B|A)*P(A) / P(B) = (2/4 * 4/7) / 3/7 = 2/3 ➔ CORRECT

Bayes Theorem - Example

- Given:
  - A doctor knows that meningitis causes stiff neck 50% of the time, so P(S|M) = 0.5
  - Prior probability of any patient having meningitis is 1/50,000 and then P(M) = 1/50,000
  - Prior probability of any patient having stiff neck is 1/20 P(S) = 1/20

- If a patient has stiff neck, what's the probability he/she has meningitis? **P(M|S) ?**

$$\mathbf{P(M|S)} = \frac{\mathbf{P(S|M)\,P(M)}}{\mathbf{P(S)}} = \frac{\mathbf{0.5 \times 1/50000}}{\mathbf{1/20}} = \mathbf{0.0002}$$

Bayesian learning

- In machine learning, we try to determine the ***best hypothesis*** from some hypothesis space H, given the observed training data D.
- In Bayesian learning, the ***best hypothesis*** means the ***most probable*** hypothesis, given the data D plus any initial knowledge about the prior probabilities of the various hypotheses in H.
- Bayes theorem provides a way to calculate the probability of a hypothesis based on prior probability, the probabilities of observing various data given the hypothesis, and the observed data itself.

Bayesian learning

P(h) is ***prior probability of hypothesis h***

- P(h) to denote the initial probability that hypothesis h holds, before observing training data.
- P(h) may reflect any background knowledge we have about the chance that h is correct.
  - If we have no such prior knowledge, then each candidate hypothesis might simply get the same prior probability.

P(D) is ***prior probability of training data D***

- The probability of D given no knowledge about which hypothesis holds

P(h|D) is ***posterior probability of h given D***

- P(h|D) is called the ***posterior probability*** of ***h,*** because it reflects our confidence that ***h*** holds after we have seen the training data ***D.***
- The posterior probability P(h|D) reflects the influence of the training data ***D,*** in contrast to the prior probability P(h), which is independent of D.

P(D|h) is ***posterior probability of D given h***

- The probability of observing data ***D*** given some world in which hypothesis ***h*** holds.
- Generally, we write **P(x|y)** to denote the probability of **event x** given **event y.**

Bayesian learning

- In ML problems, we are interested in the probability P(h|D) that h holds given the observed training data D.
- Bayes theorem provides a way to calculate the posterior probability P(h|D), from the prior probability P(h), together with P(D) and P(D|h).

Bayes Theorem: $$P(h|D) = \frac{P(D|h)\,P(h)}{P(D)}$$

- P(h|D) increases with P(h) and P(D|h) according to Bayes theorem.
- P(h|D) decreases as P(D) increases, because the more probable it is that D will be observed independent of h, the less evidence D provides in support of h.

Maximum A Posteriori (MAP) Hypothesis, h_{MAP}

- The learner considers some set of candidate hypotheses H and it is interested in finding the ***most probable hypothesis*** $h \in H$ given the observed data D
- Any such maximally probable hypothesis is called a ***maximum a posteriori (MAP) hypothesis*** h_{MAP}
- We can determine the MAP hypotheses by using Bayes theorem to calculate the posterior probability of each candidate hypothesis.

$$\mathbf{h_{MAP} = \underset{h \in H}{argmax}\ P(h|D)}$$

$$h_{MAP} = \underset{h \in H}{\text{argmax}}\ \frac{p(D|h)P(h)}{P(D)}$$

$$\mathbf{h_{MAP} = \underset{h \in H}{argmax}\ P(D|h)\ P(h)}$$

Example - Does patient have cancer or not?

- The test returns
  - a correct positive result in only 98% of the cases in which the disease is actually present,
  - a correct negative result in only 97% of the cases in which the disease is not present.
- Furthermore, .008 of the entire population have cancer.

P(cancer) = .008 P(notcancer) = .992
P(+|cancer) = .98 P(-|cancer) = .02
P(+|notcancer) = .03 P(-|notcancer) = .97

- A patient takes a lab test and the result comes back positive.

P(+|cancer) P(cancer) = .98 * .008 = .0078
P(+|notcancer) P(notcancer) = .03 * .992 = .0298

➔ ***h_{MAP} is notcancer***

- Since P(cancer|+) + P(notcancer|+) must be 1

P(cancer|+) = .0078 / (.0078+.0298) = .21
P(notcancer|+) = .0298 / (.0078+.0298) = .79

Maximum Likelihood (ML) Hypothesis, h_{ML}

- If we assume that every hypothesis in H is equally probable

  i.e. $P(h_i) = P(h_j)$ for all h_i and h_j in H

  We can only consider P(D|h) to find the most probable hypothesis.

- P(D|h) is often called the ***likelihood*** of the data ***D*** given ***h***
- Any hypothesis that maximizes P(D|h) is called a ***maximum likelihood (ML) hypothesis, h_{ML}.***

$$\mathbf{h_{ML} = \underset{h \in H}{argmax}\ P\ (D\ |h)}$$

Some Basic Formulas for Probabilities

Product Rule: Probability $P(A \wedge B)$ of a conjunction of two events A and B

$$P(A \wedge B) = P(A \mid B)\, P(B) = P(B \mid A)\, P(A)$$

Sum Rule: Probability $P(A \vee B)$ of a disjunction of two events A and B

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

Theorem of Total Probability:

If events $A_1,\ldots,A_n$ are mutually exclusive with $\sum_{i=1}^{n} P(A_i) = 1$

$$P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$$

Brute-Force Bayes Concept Learning

- A Concept-Learning algorithm considers a finite hypothesis space **H** defined over an instance space **X**
- The task is to learn the target concept (a function) **c : X → {0,1}.**
- The learner gets a set of training examples **($<x_1,d_l>$ . . . $<x_m,d_m>$)** where $\mathbf{x_i}$ is an instance from **X** and $\mathbf{d_i}$ is its target value (i.e. $c(x_i) = d_i$).

- *Brute-Force Bayes Concept Learning Algorithm* finds the maximum a posteriori hypothesis (h_{MAP}), based on Bayes theorem.

Brute-Force MAP Learning Algorithm

1. For each hypothesis h in H, calculate the posterior probability

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

2. Output the hypothesis h_{MAP} with the highest posterior probability

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}} \; P(h|D)$$

- This algorithm may require significant computation, because it applies Bayes theorem to each hypothesis in H to calculate P(h|D) .
  - While this is impractical for large hypothesis spaces,
  - The algorithm is still of interest because it provides a standard against which we may judge the performance of other concept learning algorithms.

Brute-Force MAP Learning Algorithm

- Brute-Force MAP learning algorithm must specify values for P(h) and P(D|h).
- P(h) and P(D|h) must be chosen to be consistent with the assumptions:
  1. The training data D is noise free (i.e., $d_i = c(x_i)$).
  2. The target concept c is contained in the hypothesis space H
  3. We have no a priori reason to believe that any hypothesis is more probable than any other.

- With these assumptions:

$P(h) = \frac{1}{|H|}$ for all h in H

$$A = f(x) = \begin{cases} 1, & if\ d_i = \mathrm{h}(x_i) \text{ for } all\ d_i\ in\ D \\ 0, & otherwise \end{cases}$$

Brute-Force MAP Learning Algorithm

- So, the values of P(h|D) will be:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} = \frac{0.P(h)}{P(D)} = 0$$ if h is inconsistent with D

$$P(h|D) = \frac{1.\frac{1}{|H|}}{\frac{|VS_{H,D}|}{|H|}} = \frac{1}{|VS_{H,D}|}$$ if h is consistent with D

where $VS_{H,D}$ is the version space of H with respect to D.

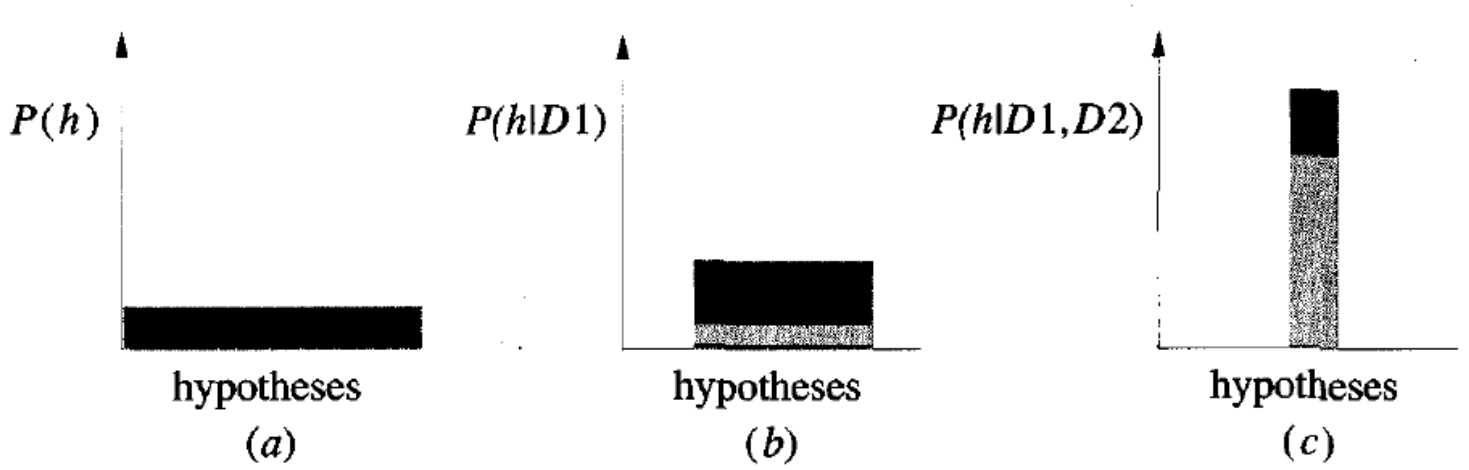
- P(D) = $|VS_{H,D}|$ / |H| because
  - the sum over all hypotheses of P(h|D) must be one and the number of hypotheses from H consistent with D is $|VS_{H,D}|$, or
  - we can derive P(D) from *the theorem of total probability* and the fact that the hypotheses are mutually exclusive (i.e., $(\forall i \neq j)(P(h_i \wedge h_j) = 0)$

$$P(D) = \sum_{h_i} P(D|h_i)P(h_i) = \sum_{h_i \in VS_{H,D}} 1 \cdot \frac{1}{|H|} + \sum_{h_i \notin VS_{H,D}} 0 \cdot \frac{0}{|H|} = \sum_{h_i \in VS_{H,D}} 1 \cdot \frac{1}{|H|} = \frac{|VS_{H,D}|}{|H|}$$

Evolution of posterior probabilities P(h|D) with increasing training data.

- Uniform prior probabilities assign equal probability to each hypothesis without considering any example in the data set.

- As training data increases to D1, then to D1 $\wedge$ D2 ,
  - the posterior probability of inconsistent hypotheses becomes zero,
  - while posterior probabilities increase for hypotheses remaining in the version space.



Bayes Optimal Classifier

- Normally we consider:
  - What is the most probable ***hypothesis*** given the training data?
- We can also consider:
  - What is the most probable ***classification*** of the new instance given the training data?

- Consider a hypothesis space containing three hypotheses, h1, h2, and h3.
  - Suppose that the posterior probabilities of these hypotheses given the training data are .4, .3, and .3 respectively.
  - Thus, h1 is the MAP hypothesis.
  - Suppose a new instance x is encountered, which is classified positive by *h1,* but negative by *h2* and h3.
  - Taking all hypotheses into account, the probability that x is positive is .4 (the probability associated with *h1),* and the probability that it is negative is .6.
  - The most probable classification (negative) in this case is different from the classification generated by the MAP hypothesis.

Bayes Optimal Classifier

- The most probable classification of the new instance is obtained by combining the predictions of all hypotheses, weighted by their posterior probabilities.
- If the possible classification of the new example can take on any value v_j from some set **V**, then the probability **P(v_j | D)** that the correct classification for the new instance is v_j :

$$P(v_j|D) = \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

- **Bayes optimal classification:**

$$argmax_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

Bayes Optimal Classifier - Example

$P(h_1|D) = .4 \quad P(-|h_1) = 0 \quad P(+|h_1) = 1$

$P(h_2|D) = .3 \quad P(-|h_2) = 1 \quad P(+|h_2) = 0$

$P(h_3|D) = .3 \quad P(-|h_3) = 1 \quad P(+|h_3) = 0$

Probabilities:

$$\sum_{h_i \in H} P(+|h_i)P(h_i|D) = .4$$

$$\sum_{h_i \in H} P(-|h_i)P(h_i|D) = .6$$

Result:

$$argmax_{v_j \in \{+,-\}} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) \quad \rightarrow \quad -$$

Bayes Optimal Classifier

- Although the Bayes optimal classifier obtains the best performance that can be achieved from the given training data, it can be quite costly to apply.
  - The expense is due to the fact that it computes the posterior probability for every hypothesis in H and then combines the predictions of each hypothesis to classify each new instance.

Naive Bayes Classifier

- One highly practical Bayesian learning method is Naive Bayes Learner (***Naive Bayes Classifier).***
- The naive Bayes classifier applies to learning tasks where each instance ***x*** is described by a conjunction of attribute values and where the target function f(x) can take on any value from some finite set V.
- A set of training examples is provided, and a new instance is presented, described by the tuple of attribute values **(a_l, a_2 ...a_n).**
- The learner is asked to predict the target value (classification), for this new instance.

Naive Bayes Classifier

- The Bayesian approach to classifying the new instance is to assign the most probable target value v_{MAP}, given the attribute values (a_l, a_2 ... a_n) that describe the instance.

$$\mathbf{v_{MAP}} = \underset{\mathbf{v_j} \in \mathbf{V}}{\mathbf{argmax}}\, \mathbf{P}(\mathbf{v_j}|\mathbf{a_1}, \ldots, \mathbf{a_n})$$

- By Bayes theorem:

$$\mathbf{v_{MAP}} = \underset{\mathbf{v_j} \in \mathbf{V}}{\mathbf{argmax}}\, \frac{\mathbf{P}(\mathbf{a_1},\ldots,\mathbf{a_n}|\mathbf{v_j})\, \mathbf{P}(\mathbf{v_j})}{\mathbf{P}(\mathbf{a_1},\ldots,\mathbf{a_n})}$$

$$\mathbf{v_{MAP}} = \underset{\mathbf{v_j} \in \mathbf{V}}{\mathbf{argmax}}\, \mathbf{P}(\mathbf{a_1}, \ldots, \mathbf{a_n}|\mathbf{v_j})\mathbf{P}(\mathbf{v_j})$$

Naive Bayes Classifier

- It is easy to estimate each of the $\mathbf{P(v_j)}$ simply by counting the frequency with which each target value $\mathbf{v_j}$ occurs in the training data.
- However, estimating the different $\mathbf{P(a_1,a_2 \ldots a_n \mid v_j)}$ terms is not feasible unless we have a very, very large set of training data.
  - The problem is that the number of these terms is equal to the number of possible instances times the number of possible target values.
  - Therefore, we need to see every instance in the instance space many times in order to obtain reliable estimates.

Naive Bayes Classifier

- The naive Bayes classifier is based on the simplifying assumption that the attribute values are ***conditionally independent*** given the target value.
- For a given the target value of the instance, the probability of observing conjunction $\mathbf{a_1,a_2 \ldots a_n}$, is just the product of the probabilities for the individual attributes:

$$P(a_1, \ldots, a_n | v_j) = \prod_i P(a_i | v_j)$$

Naive Bayes classifier:

$$V_{NB} = argmax_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$$

Naive Bayes Classifier: Independence of Events

The events A and B are **INDEPENDENT**

if and only if **P(A,B) = P(A)*P(B)**

Example: Bit strings of length 3 is {000,001,010,011,100,101,110,111}

Event A: A randomly generated bit string of length three begins with a 1.

Event B: A randomly generated bit string of length three ends with a 1.

P(A) = 4/8 100,101,110,111 P(B) = 4/8 001,011,101,111

P(A,B) = 2/8 101,111 Are A and B independent?

P(A)*P(B) = (4/8) * (4/8) = 16/64 = 2/8 = P(A,B)

➔ A and B are independent.

Event C: A randomly generated bit string of length three contains with two 1s.

P(C) = 3/8 011,101,110

P(A,C) = 2/8 101,110 Are A and C independent?

P(A)*P(C) = (4/8)*(3/8) = 12/64 = 3/16 $\neq$ 2/8

➔ A and C are NOT independent.

Naive Bayes Classifier - Example

Day	Outlook	Temp.	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	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	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

Naive Bayes Classifier – Example

- New instance to classify:

  (Outlook=sunny, Temperature=cool, Humidity=high, Wind=strong)

- Our task is to predict the target value (yes or no) of the target concept *PlayTennis* for this new instance.

$$V_{NB} = argmax_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$$

$$\mathbf{v_{NB}} = \underset{\mathbf{v_j} \in \{\mathbf{yes}, \mathbf{no}\}}{\mathbf{argmax}} \; \mathbf{P}(\mathbf{v_j}) \; \mathbf{P}(\mathbf{Outlook} = \mathbf{sunny} | \mathbf{v_j}) \, \mathbf{P}(\mathbf{Temperature} = \mathbf{cool} | \mathbf{v_j}) \, \mathbf{P}(\mathbf{Humidity} = \mathbf{high} | \mathbf{v_j}) \, \mathbf{P}(\mathbf{Wind} = \mathbf{strong} | \mathbf{v_j})$$

Naive Bayes Classifier - Example

- P(PlayTennis = yes) = 9/14 = .64
- P(PlayTennis = no) = 5/14 = .36

$$\mathbf{P(yes)\ P(sunny|yes)P(cool|yes)P(high|yes)P(strong|yes)} = \mathbf{.0053}$$

$$\mathbf{P(no)\ P(sunny|no)P(cool|no)P(high|no)P(strong|no)} = \mathbf{.0206}$$

➔ Thus, the naive Bayes classifier assigns the target value ***PlayTennis*** = ***no*** to this new instance, based on the probability estimates learned from the training data.

- Furthermore, by normalizing the above quantities to sum to one we can calculate the ***conditional probability*** that the target value is ***no,*** given the observed attribute values.

.0206 / (.0206 + .0053) = .795

Estimating Probabilities

- **P(Wind=strong | PlayTennis=no)** by the fraction $\mathbf{n_c/n}$ where ***n*** = 5 is the total number of training examples for which **PlayTennis=no,** and n_c = 3 is the number of these for which **Wind=strong.**
- When n_c is zero
  - $\mathbf{n_c/n}$ will be zero too
  - this probability term will dominate
- To avoid this difficulty we can adopt a Bayesian approach to estimating the probability, using the m-estimate defined as follows.

  m-estimate of probability: $(n_c + m*p) / (n + m)$
- if an attribute has ***k*** possible values we set p = 1/k .
  - p=0.5 because Wind has two possible values.
- m is called the equivalent sample size
  - augmenting the ***n*** actual observations by an additional m virtual samples distributed according to p.

Avoiding the Zero-Probability Problem

- Naïve Bayesian prediction requires each conditional probability to be a **non-zero value**. Otherwise, the predicted probability will be zero

$$P(x_1, x_2, \ldots, x_n \mid C_i) = P(x_1 \mid C_i) * P(x_2 \mid C_i) * \ldots * P(x_n \mid C_i)$$

- In order to avoid zero probability values, we apply smoothing techniques.
- One of these smoothing techniques is **add-one smoothing**.

- $P(A=v1 \mid C_i) = N_{v1Ci} / N_{ci}$
- $P(A=v2 \mid C_i) = N_{v2Ci} / N_{ci}$
- $P(A=v3 \mid C_i) = N_{v3Ci} / N_{ci}$

$P(A=v1 \mid C_i) = (N_{v1Ci} + 1) / (N_{ci}+3)$

$P(A=v2 \mid C_i) = (N_{v2Ci} + 1) / (N_{ci}+3)$

$P(A=v3 \mid C_i) = (N_{v3Ci} + 1) / (N_{ci}+3)$

Naive Bayes Classifier - Example

Dataset has 14 tuples.

Two classes:

buyscomputer=yes

buyscomputer=no

P(bc=yes) = 9/14

P(bc=no) = 5/14

age	income	student	creditrating	buyscomputer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Naive Bayes Classifier – Example

Computing Probabilities from Training Dataset

P(age=b31|bc=yes)=2/9
P(age=i31|bc=yes)=4/9
P(age=g40|bc=yes)=3/9

P(age=b31|bc=no)=3/5
P(age=i31|bc=no)=0
P(age=g40|bc=no)=2/5

P(inc=high|bc=yes)=2/9
P(inc=med|bc=yes)=4/9
P(inc=low|bc=yes)=3/9

P(inc=high|bc=no)=2/5
P(inc=med|bc=no)=2/5
P(inc=low|bc=no)=1/5

P(std=yes|bc=yes)=6/9
P(std=no|bc=yes)=3/9

P(std=yes|bc=no)=1/5
P(std=no|bc=no)=4/5

P(cr=exc|bc=yes)=3/9
P(cr=fair|bc=yes)=6/9

P(cr=exc|bc=no)=3/5
P(cr=fair|bc=no)=2/5

age	income	student	creditrating	buyscomputer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

P(bc=yes) = 9/14

P(bc=no) = 5/14

Naive Bayes Classifier – Example

Finding Classification

X: (age <= 30 , income = medium, student = yes, creditrating = fair)

P(X|bc=yes)=P(age=b30|bc=yes)*P(inc=med|bc=yes)*P(std=yes|bc=yes)*P(cr=fair|bc=yes)
= 2/9 * 4/9 * 6/9 * 6/9 = 0.044

P(X|bc=no)=P(age=b30|bc=no)*P(inc=med|bc=no)*P(std=yes|bc=no)*P(cr=fair|bc=no)
= 3/5 * 2/5 * 1/5 * 2/5 = 0.019

P(X|bc=yes)*P(bc=yes) = 0.044 * 9/14 = 0.028

P(X|bc=no)*P(bc=no) = 0.019 * 5/14 = 0.007

➔ Therefore, X belongs to class "buyscomputer = yes"

Confidence of the classification: 0.028/(0.028+0.007) = 0.80 80%