# Naive Bayes

**Thanh-Nghi Do**

**Can Tho University**

***dtnghi@cit.ctu.edu.vn***

**Can Tho**

**Dec. 2019**

# Content

- Introduction
- Naive Bayes
- Conclusion

# Content

- **<u>Introduction</u>**
- Naive Bayes
- Conclusion

# Bayesian classification

- Machine learning family
  - Bayes rules
  - Conditional independence
  - Naive Bayes and Bayesian Networks
  - Classification, clustering, etc.
  - Data analysis, text categorization, spam classification, etc.

# Top 10 Data Mining Algorithms

(Kdnuggets)


Here are the algorithms:

- 1. C4.5
- 2. k-means
- 3. Support vector machines
- 4. Apriori
- 5. EM
- 6. PageRank
- 7. AdaBoost
- 8. kNN
- 9. Naive Bayes
- 10. CART

# Content

- Introduction
- **Naive Bayes**
- Conclusion

# Naive Bayes

- Naive
  - Features are equally important
  - Assumption that $X_i$ are conditionally independent, given $Y$
    $P(X_1|X_2, Y) = P(X_1|Y)$

- Comments
  - Independence assumption is almost never correct!
  - But … this scheme works well in practice ☺

# Training dataset Weather [Outlook, Temp, Humidity, Windy] → Play

| 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 |
| 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 |

# Training dataset Weather [Outlook, Temp, Humidity, Windy] → Play

| Outlook | Yes | No | Temperature | Yes | No | Humidity | Yes | No | Windy | Yes | No | Play Yes | Play No |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sunny | 2 | 3 | Hot | 2 | 2 | High | 3 | 4 | False | 6 | 2 | 9 | 5 |
| Overcast | 4 | 0 | Mild | 4 | 2 | Normal | 6 | 1 | True | 3 | 3 | | |
| Rainy | 3 | 2 | Cool | 3 | 1 | | | | | | | | |
| Sunny | 2/9 | 3/5 | Hot | 2/9 | 2/5 | High | 3/9 | 4/5 | False | 6/9 | 2/5 | 9/14 | 5/14 |
| Overcast | 4/9 | 0/5 | Mild | 4/9 | 2/5 | Normal | 6/9 | 1/5 | True | 3/9 | 3/5 | | |
| Rainy | 3/9 | 2/5 | Cool | 3/9 | 1/5 | | | | | | | | |

| 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 |
| 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 |

# Training dataset Weather [Outlook, Temp, Humidity, Windy] → Play

| Outlook | | | Temperature | | | Humidity | | | Windy | | | Play | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *Yes* | *No* | | *Yes* | *No* | | *Yes* | *No* | | *Yes* | *No* | *Yes* | *No* |
| Sunny | 2 | 3 | Hot | 2 | 2 | High | 3 | 4 | False | 6 | 2 | 9 | 5 |
| Overcast | 4 | 0 | Mild | 4 | 2 | Normal | 6 | 1 | True | 3 | 3 | | |
| Rainy | 3 | 2 | Cool | 3 | 1 | | | | | | | | |
| Sunny | 2/9 | 3/5 | Hot | 2/9 | 2/5 | High | 3/9 | 4/5 | False | 6/9 | 2/5 | 9/14 | 5/14 |
| Overcast | 4/9 | 0/5 | Mild | 4/9 | 2/5 | Normal | 6/9 | 1/5 | True | 3/9 | 3/5 | | |
| Rainy | 3/9 | 2/5 | Cool | 3/9 | 1/5 | | | | | | | | |

- Play = yes/no?

| Outlook | Temp. | Humidity | Windy | Play |
|---|---|---|---|---|
| Sunny | Cool | High | True | ? |

# Training dataset Weather [Outlook, Temp, Humidity, Windy] → Play

| Outlook | | | Temperature | | | Humidity | | | Windy | | | Play | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *Yes* | *No* | | *Yes* | *No* | | *Yes* | *No* | | *Yes* | *No* | *Yes* | *No* |
| Sunny | 2 | 3 | Hot | 2 | 2 | High | 3 | 4 | False | 6 | 2 | 9 | 5 |
| Overcast | 4 | 0 | Mild | 4 | 2 | Normal | 6 | 1 | True | 3 | 3 | | |
| Rainy | 3 | 2 | Cool | 3 | 1 | | | | | | | | |
| Sunny | 2/9 | 3/5 | Hot | 2/9 | 2/5 | High | 3/9 | 4/5 | False | 6/9 | 2/5 | 9/14 | 5/14 |
| Overcast | 4/9 | 0/5 | Mild | 4/9 | 2/5 | Normal | 6/9 | 1/5 | True | 3/9 | 3/5 | | |
| Rainy | 3/9 | 2/5 | Cool | 3/9 | 1/5 | | | | | | | | |

| Outlook | Temp. | Humidity | Windy | Play |
|---|---|---|---|---|
| Sunny | Cool | High | True | ? |

- Play = yes/no?

Likelihood of the two classes

For "yes" = 2/9 × 3/9 × 3/9 × 3/9 × 9/14 = 0.0053

For "no" = 3/5 × 1/5 × 4/5 × 3/5 × 5/14 = 0.0206

Conversion into a probability by normalization:

P("yes") = 0.0053 / (0.0053 + 0.0206) = 0.205

P("no") = 0.0206 / (0.0053 + 0.0206) = 0.795

# Bayes rule

- Probability of event $H$ given evidence $E$ :

$$\Pr[H \mid E] = \frac{\Pr[E \mid H]\Pr[H]}{\Pr[E]}$$

- *A priori* probability of $H$ : $\Pr[H]$
  - Probability of event *before* evidence is seen
- *A posteriori* probability of $H$ : $\Pr[H \mid E]$
  - Probability of event *after* evidence is seen

# Naive Bayes for classification

- Classification learning: what's the probability of the class given an example?
  - Evidence $E$ = example
  - Event $H$ = class value for example
- Naïve assumption: evidence splits into parts (i.e. features) that are independent

$$\Pr[H \mid E] = \frac{\Pr[E_1 \mid H]\Pr[E_2 \mid H]\ldots\Pr[E_n \mid H]\Pr[H]}{\Pr[E]}$$

# Naive Bayes for classification

| Outlook | Temp. | Humidity | Windy | Play |
|---|---|---|---|---|
| Sunny | Cool | High | True | ? |

***Evidence E***

***Probability of class "yes"***

$$\begin{aligned}\Pr[yes \mid E] = {} & \Pr[Outlook = Sunny \mid yes] \\ & \times \Pr[Temperature = Cool \mid yes] \\ & \times \Pr[Humidity = High \mid yes] \\ & \times \Pr[Windy = True \mid yes] \\ & \times \frac{\Pr[yes]}{\Pr[E]}\end{aligned}$$

$$= \frac{\frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14}}{\Pr[E]}$$

# The "zero-frequency problem"

- What if a feature value doesn't occur with every class value? (e.g. "Humidity = high" for class "yes")
  - Probability will be zero! $\Pr[Humidity = High \mid yes] = 0$
  - *A posteriori* probability will also be zero! $\Pr[yes \mid E] = 0$

- Remedy: add 1 to the count for every attribute value-class combination (Laplace estimator)
- Result: probabilities will never be zero!
(also: stabilizes probability estimates)

# Laplace estimator

- Example: feature outlook for class yes

$$\frac{2+\mu/3}{9+\mu} \qquad \frac{4+\mu/3}{9+\mu} \qquad \frac{3+\mu/3}{9+\mu}$$

***Sunny*** ***Overcast*** ***Rainy***

- Weights don't need to be equal (but they must sum to 1)

$$\frac{2+\mu p_1}{9+\mu} \qquad \frac{4+\mu p_2}{9+\mu} \qquad \frac{3+\mu p_3}{9+\mu}$$

***Sunny*** ***Overcast*** ***Rainy***

# Missing values

- Training: example is not included in frequency count for attribute value-class combination
- Classification: feature will be omitted from calculation
- Example:

| Outlook | Temp. | Humidity | Windy | Play |
|---|---|---|---|---|
| ? | Cool | High | True | ? |

Likelihood of yes = $3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0238$

Likelihood of no = $1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0343$

P(yes) = 0.0238 / (0.0238 + 0.0343) = 41

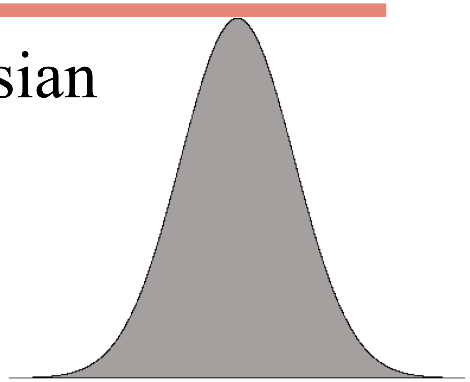
P(no) = 0.0343 / (0.0238 + 0.0343) = 59

# Numerical features

| Outlook | Temperature | Humidity | Windy | Play |
|---|---|---|---|---|
| sunny | 85 | 85 | false | no |
| sunny | 80 | 90 | true | no |
| overcast | 83 | 78 | false | yes |
| rain | 70 | 96 | false | yes |
| rain | 68 | 80 | false | yes |
| rain | 65 | 70 | true | no |
| overcast | 64 | 65 | true | yes |
| sunny | 72 | 95 | false | no |
| sunny | 69 | 70 | false | yes |
| rain | 75 | 80 | false | yes |
| sunny | 75 | 70 | true | yes |
| overcast | 72 | 90 | true | yes |
| overcast | 81 | 75 | false | yes |
| rain | 71 | 80 | true | no |

# Numerical features

- Usual assumption: features have a normal or Gaussian probability distribution (given the class)
- The probability density function for the normal distribution is defined by two parameters:
  - *Mean:* $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$
  - *Standard deviation:* $\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2$
  - The density function: $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$



Karl Gauss, 1777-1855
great German mathematician

# Numerical features

| Outlook | | | Temperature | | | Humidity | | | Windy | | | Play | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *yes* | *no* | | *yes* | *no* | | *yes* | *no* | | *yes* | *no* | *yes* | *no* |
| sunny | 2 | 3 | | 83 | 85 | | 86 | 85 | false | 6 | 2 | 9 | 5 |
| overcast | 4 | 0 | | 70 | 80 | | 96 | 90 | true | 3 | 3 | | |
| rainy | 3 | 2 | | 68 | 65 | | 80 | 70 | | | | | |
| | | | | 64 | 72 | | 65 | 95 | | | | | |
| | | | | 69 | 71 | | 70 | 91 | | | | | |
| | | | | 75 | | | 80 | | | | | | |
| | | | | 75 | | | 70 | | | | | | |
| | | | | 72 | | | 90 | | | | | | |
| | | | | 81 | | | 75 | | | | | | |
| sunny | 2/9 | 3/5 | *mean* | 73 | 74.6 | *mean* | 79.1 | 86.2 | false | 6/9 | 2/5 | 9/14 | 5/14 |
| overcast | 4/9 | 0/5 | *std. dev.* | 6.2 | 7.9 | *std. dev.* | 10.2 | 9.7 | true | 3/9 | 3/5 | | |
| rainy | 3/9 | 2/5 | | | | | | | | | | | |

$$f(temperature = 66 \mid yes) = \frac{1}{\sqrt{2\pi}6.2} e^{-\frac{(66-73)^2}{2*6.2^2}} = 0.0340$$

# Numerical features

- Play = yes/no?

| Outlook | Temp. | Humidity | Windy | Play |
|---|---|---|---|---|
| Sunny | 66 | 90 | true | ? |

Likelihood of yes = $2/9 \times 0.0340 \times 0.0221 \times 3/9 \times 9/14 = 0.000036$

Likelihood of no = $3/5 \times 0.0291 \times 0.0380 \times 3/5 \times 5/14 = 0.000136$

P(yes) = 0.000036 / (0.000036 + 0. 000136) = 20.9

P(no) = 0.000136 / (0.000036 + 0. 000136) = 79.1

# Content

- Introduction
- Naive Bayes
- **Conclusion**

# Conclusions

- Naïve Bayes
  - Bayes rule
  - Works surprisingly well (even if independence assumption is clearly violated)
  - Why? Because classification doesn't require accurate probability estimates as long as maximum probability is assigned to correct class
  - However: adding too many redundant features will cause problems (e.g. identical features)
  - Note also: many numeric features are not normally distributed → kernel density estimators

# Extensions

- Improvements
  - Select best features (e.g. with greedy search)
  - Often works as well or better with just a fraction of all features
  - Bayesian Networks

Merci!