Naïve Bayes Model

Classification Problem Revisited

We want to determine P(C|X), that is, the probability that the record $X=<x_1,\ldots,x_k>$ is of class C.

e.g. X = <rain, hot, high, light> PlayTennis?

Day	Outlook	Temperature	Humidity	Wind	*PlayTennis?*
1	Sunny	Hot	High	Light	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Light	Yes
4	Rain	Mild	High	Light	Yes
5	Rain	Cool	Normal	Light	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Light	No
9	Sunny	Cool	Normal	Light	Yes
10	Rain	Mild	Normal	Light	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Light	Yes
14	Rain	Mild	High	Strong	No

Bayes Model

- Treat each attribute and class label as random variables.

- Given a sample **x** with attributes ($x_1, x_2, \ldots, x_n$):
  - Goal is to predict class *C*.
  - Specifically, we want to find the value of C_i that maximizes $p(C_i \mid x_1, x_2, \ldots, x_n)$.

- Can we estimate $p(C_i \mid x_1, x_2, \ldots, x_n)$ directly from data?

Bayes Model

- Compute the posterior probability $p(C_i \mid x_1, x_2, \ldots, x_n)$ for each value of C_i using Bayes theorem:

$$p(C_i \mid x_1, x_2, \ldots, x_n) = \frac{p(x_1, x_2, \ldots, x_n \mid C_i)\, p(C_i)}{p(x_1, x_2, \ldots, x_n)}$$

- Choose value of C_i that maximizes

  $p(C_i \mid x_1, x_2, \ldots, x_n)$

- Equivalent to choosing value of C_i that maximizes

  $p(x_1, x_2, \ldots, x_n \mid C_i)\, p(C_i)$

  (*We can ignore denominator – why?*)

- Easy to estimate priors $p(C_i)$ from data. (*How?*)
- The real challenge: how to estimate $p(x_1, x_2, \ldots, x_n \mid C_i)$?

Bayes Model

- How to estimate $p(x_1, x_2, \ldots , x_n \mid C_i)$?

- In the general case, where the attributes x_j have dependencies, this requires estimating the full joint distribution $p(x_1, x_2, \ldots , x_n)$ for each class C_i.

Naïve Bayes Model

- Assume independence among attributes x_j when class is given:

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

- Usually straightforward and practical to estimate $p(x_j \mid C_i)$ for all x_j and C_i.

Model Learning: Parameter Estimation

Objective Function

We don’t have any objective function for parameter learning. Instead we have to compute all the prior and likelihood probabilites.

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

The learning phase for tennis example

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

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

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

Conditional probability tables corresponding to four variables

outlook	
P(sunny|p) = 2/9	P(sunny|n) = 3/5
P(overcast|p) = 4/9	P(overcast|n) = 0
P(rain|p) = 3/9	P(rain|n) = 2/5
temperature	
P(hot|p) = 2/9	P(hot|n) = 2/5
P(mild|p) = 4/9	P(mild|n) = 2/5
P(cool|p) = 3/9	P(cool|n) = 1/5
humidity	
P(high|p) = 3/9	P(high|n) = 4/5
P(normal|p) = 6/9	P(normal|n) = 2/5
windy	
P(strong|p) = 3/9	P(strong|n) = 3/5
P(light|p) = 6/9	P(light|n) = 2/5

Play-tennis example: classifying X

- **An unseen sample**

X = <rain, hot, high, light>

P(play | X) ⇨ P(X | play) · P(play) =

=P(rain | play) · P(hot | play) ·

· P(high | play) · P(light | play) · P(p) =

= 3/9 ·2/9 · 3/9 · 6/9 · 9/14 =

= 0.010582

outlook	
P(sunny|p) = 2/9	P(sunny|n) = 3/5
P(overcast|p) = 4/9	P(overcast|n) = 0
P(rain|p) = 3/9	P(rain|n) = 2/5
temperature	
P(hot|p) = 2/9	P(hot|n) = 2/5
P(mild|p) = 4/9	P(mild|n) = 2/5
P(cool|p) = 3/9	P(cool|n) = 1/5
humidity	
P(high|p) = 3/9	P(high|n) = 4/5
P(normal|p) = 6/9	P(normal|n) = 2/5
windy	
P(strong|p) = 3/9	P(strong|n) = 3/5
P(light|p) = 6/9	P(light|n) = 2/5

Play-tennis example: classifying X

P(don't play | X) ⇨ P(X | don't play) · P(don't play) =

= P(rain | don't play) · P(hot | don't play) ·

· P(high | don't play) · P(light | don't play) ·

· P(don't play) =

= 2/5 · 2/5 · 4/5 · 2/5 · 5/14 = 0.018286 > 0.010582

Sample X is classified in class n (don't play)

Naïve Bayes Algorithm: Continuous Inputs

- Conditional probability often modeled with gaussian distribution.
- Generate a separate gaussian model for each class separately.

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

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 $C = c_i$

Example

- 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)

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

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

Issues in Naïve Bayes Implementation

Correlated Features

Violation of Independence Assumption

- For many real world tasks, $P(X_1,\cdots,X_n|C) \neq P(X_1|C)\cdots P(X_n|C)$
- Nevertheless, naïve Bayes works surprisingly well anyway!

Zero Frequency Problem

- What if an attribute 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!
    (No matter how likely the other values are!)

$$\Pr[yes \mid <\ldots, hum = high>] = 0$$

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

Zero Frequency Problem

- In some cases adding a constant different from 1 might be more appropriate
- Example: attribute *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)

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

Missing Data Problem

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

outlook	
P(sunny|p) = 2/9	P(sunny|n) = 3/5
P(overcast|p) = 4/9	P(overcast|n) = 0
P(rain|p) = 3/9	P(rain|n) = 2/5
temperature	
P(hot|p) = 2/9	P(hot|n) = 2/5
P(mild|p) = 4/9	P(mild|n) = 2/5
P(cool|p) = 3/9	P(cool|n) = 1/5
humidity	
P(high|p) = 3/9	P(high|n) = 4/5
P(normal|p) = 6/9	P(normal|n) = 2/5
windy	
P(strong|p) = 3/9	P(strong|n) = 3/5
P(light|p) = 6/9	P(light|n) = 2/5

Example:

Outlook	Temp.	Humidity	Windy	Play
?	Cool	High	Strong	?

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%

Floating point underflow

- Multiplying lots of probabilities can result in floating-point underflow.
- Since $\log(xy) = \log(x) + \log(y)$
  - Better to sum logs of probabilities instead of multiplying probabilities.

Pros & Cons of Naïve Bayes Model

Pros of NB Model

- Fast Learning and prediction
  - 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 estimated distributions
- Performance competitive to most of state-of-the-art classifiers even in presence of violating independence assumption
- Many successful applications, e.g., spam mail filtering
- A good candidate of a base learner in ensemble learning

Cons of NB Model

- *NOT* robust to redundant attributes.
  - Independence assumption does not hold in this case.
  - Use other techniques such as Bayesian Belief Networks (BBN).