The Naïve Bayes Classifier

Lecture 10

Outline

1. Bayes' Rule
2. Learning via probability estimates
3. Feasibility via conditional independence
4. Estimating likelihoods
   – Multinomial with smoothing
   – Gaussian
5. Practical Issues

Axiom of Conditional Probability

Conditional Probability

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

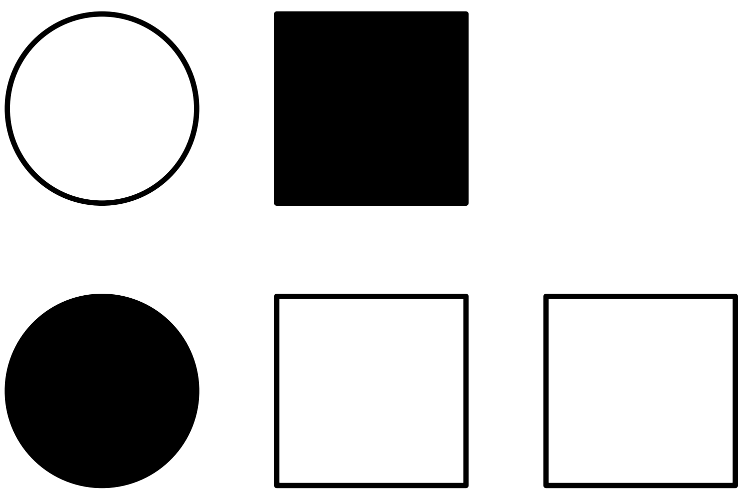
Joint Probability

Simple Example

- A = filled
- B = shape is square

$$P(A) = \frac{2}{5} \qquad P(B) = \frac{3}{5}$$

$$P(A|B) = \frac{1}{3} \qquad P(B|A) = \frac{1}{2}$$



$$P(A,B) =$$

$$=$$

$$= \frac{1}{5}$$

Bayes' Rule

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

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

$$\underbrace{P(A|B)}_{\text{Posterior}} = \frac{\overbrace{P(B|A)}^{\text{Likelihood}} \cdot \overbrace{P(A)}^{\text{Prior}}}{\underbrace{P(B)}_{\text{Evidence/Support}}}$$

Why Does Bayes' Rule Matter?

Often we know/can estimate likelihood and prior information easier than the posterior

$$P(\text{Hypothesis}|\text{Data}) = \frac{P(\text{Data}|\text{Hypothesis}) \cdot P(\text{Hypothesis})}{P(\text{Data})}$$

<u>Clinical example</u>

- A: person has cancer
- B: person smokes

Easy from historical data

- P(A) = 10%
- P(B) = 40%
- P(B|A) = 80%

$$P(A|B) = 20\%$$

Learning via Probability Estimates

- Consider the posterior probability distribution over a discrete set of classes (*C*) and fixed set of features (***x***; each continuous or discrete)

$$P(C_k|\boldsymbol{x}) = \frac{P(C_k) \cdot P(\boldsymbol{x}|C_k)}{P(\boldsymbol{x})}$$

- The maximum a posteriori (MAP) decision rule says to select the class that maximizes the posterior, thus…

$$\hat{y} = \underset{k \in \{1 \ldots K\}}{\arg\max} \frac{P(C_k) \cdot P(\boldsymbol{x}|C_k)}{P(\boldsymbol{x})}$$

Note

- The evidence term is only dependent on the data, and applies a normalizing constant (i.e. p's sum to 1)

$$P(\boldsymbol{x}) = \sum_k P(\boldsymbol{x}, C_k)$$
$$= \sum_k P(\boldsymbol{x}|C_k) \cdot P(C_k)$$

- For classification we care only about selecting the maximum value, and so we can maximize the numerator and ignore the denominator

$$\hat{y} = \underset{k \in \{1 \ldots K\}}{\arg\max}\, P(C_k) P(\boldsymbol{x}|C_k)$$

How Much Data is Necessary?

$$\hat{y} = \underset{k \in \{1 \ldots K\}}{\arg\max}\, P(C_k) P(\boldsymbol{x} | C_k)$$

- We can reasonably estimate the class prior via data (e.g. 2 classes ~ 100 points)

- However, likelihood is exponential
  - P({0,0,0…,0} | 0) x 100
  - P({0,0,0…,0} | 1) x 100
  - P({0,0,0…,1} | 0) x 100
  - P({0,0,0…,1} | 1) x 100

  …

Feasibility via Conditional Independence

- The term **naïve** refers to the algorithmic assumption that each feature is conditionally independent of *every other* feature
  - This has the effect of reducing the necessary estimation data from exponential to linear

- In practice, while the independence assumption typically may not hold, Naïve Bayes works surprisingly well and is efficient for very large data sets with many features

Conditional Independence

X is conditionally independent of Y given Z, if and only if the probability distribution governing X is independent of the value of Y given Z

$$(\forall i, j, k) P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

Deriving Naïve Bayes

Consider the two-feature example:

$$
\begin{aligned}
P(X, Y) &= P(X_1, X_2 | Y) \\
&= P(X_1 | X_2, Y) \cdot P(X_2 | Y)
\end{aligned}
$$

Now apply the conditional independence assumption…

$$
= P(X_1 | Y) \cdot P(X_2 | Y)
$$

More Generally…

$$
\begin{aligned}
P(X_1,\ldots,X_n|C_k) &= P(X_1|C_k)\cdot P(X_2,\ldots,X_n|C_k,X_1)\\
&= P(X_1|C_k)\cdot P(X_2|C_k,X_1)\cdot P(X_3,\ldots,X_n|C_k,X_1,X_2)\\
&= \ldots
\end{aligned}
$$

where…

$$
\begin{aligned}
P(X_i|C_k,X_j) &= P(X_i|C_k)\\
P(X_i|C_k,X_j,X_q) &= P(X_i|C_k)\\
P(X_i|C_k,X_j,X_q,\ldots) &= P(X_i|C_k)
\end{aligned}
$$

And so…

$$\begin{aligned}\hat{y} &= \arg\max_{k \in \{1 \ldots K\}} P(C_k) \cdot P(\boldsymbol{x}|C_k) \\ &= \arg\max_{k \in \{1 \ldots K\}} P(C_k) \cdot \prod_{i=1}^{n} P(x_i|C_k)\end{aligned}$$

Parameter Estimation – Prior

- Default approach
  - (# examples of class) / (# examples)
- Could also assume equiprobable
  - 1/(# distinct classes)

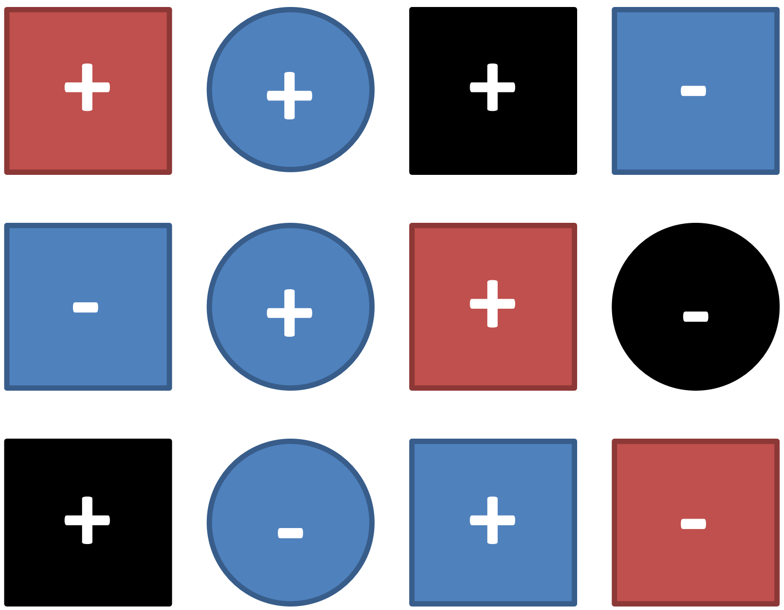
Parameter Estimation - Likelihood

- For discrete feature values, can assume a **multinomial distribution** and use the maximum likelihood estimate (MLE)

- For continuous values, a common assumption is that for each discrete class label the distribution of each continuous feature is **Gaussian**

Example

Dataset



Color = {Red, Blue, Black, Orange}

Shape = {Square, Circle}

Input

$$P(+) = \frac{7}{12} \qquad P(-) = \frac{5}{12}$$

$$P(\text{Blue}|+) = \frac{3}{7} \qquad P(\text{Blue}|-) = \frac{3}{5}$$

$$P(\text{Square}|+) = \frac{5}{7} \qquad P(\text{Square}|-) = \frac{3}{5}$$

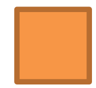
$$P(\boldsymbol{x}|+) = \frac{3}{7} \cdot \frac{5}{7} \sim 0.31$$

$$P(\boldsymbol{x}|-) = \frac{3}{5} \cdot \frac{3}{5} = 0.36$$

$$P(+|\text{Blue}, \text{Square}) = \frac{7}{12} \cdot \frac{3}{7} \cdot \frac{5}{7} \sim 0.18 \checkmark$$

$$P(-|\text{Blue}, \text{Square}) = \frac{5}{12} \cdot \frac{3}{5} \cdot \frac{3}{5} = 0.15$$

Additive Smoothing

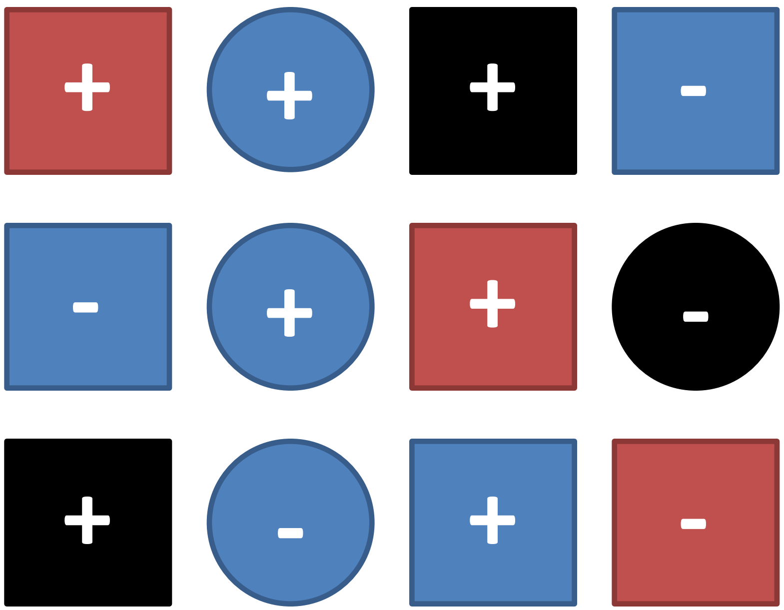
- An issue that arises in the calculation is what to do when evaluating a feature value you haven't seen (e.g. )

- To accommodate, use additive smoothing
  - d = feature dimensionality
  - α = smoothing parameter/strength (≥0)
    - 0 = no smoothing
    - <1 = Lidstone smoothing
    - ≥1 = Laplace smoothing

$$\frac{x + \alpha}{N + \alpha d}$$

Example, Laplace Smoothing

Dataset



Color = {Red, Blue, Black, Orange}

Shape = {Square, Circle}

Input

$$P(+) = \frac{7}{12} \qquad P(-) = \frac{5}{12}$$

$$P(\text{Orange}|+) = \frac{0+1}{7+4} = \frac{1}{11} \qquad P(\text{Orange}|-) = \frac{0+1}{5+4} = \frac{1}{9}$$

$$P(\text{Square}|+) = \frac{5}{7} \qquad P(\text{Square}|-) = \frac{3}{5}$$

$$P(\boldsymbol{x}|+) = \frac{1}{11} \cdot \frac{5}{7} \sim 0.06$$

$$P(\boldsymbol{x}|-) = \frac{1}{9} \cdot \frac{3}{5} \sim 0.07$$

$$P(+|\text{Orange}, \text{Square}) = \frac{7}{12} \cdot \frac{1}{11} \cdot \frac{5}{7} \sim 0.04$$

$$P(-|\text{Orange}, \text{Square}) = \frac{5}{12} \cdot \frac{1}{9} \cdot \frac{3}{5} \sim 0.03$$

Gaussian MLE Estimate

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2}$$

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

		Humidity	Mean	Std. Dev.
Play Golf	yes	86 96 80 65 70 80 70 90 75	79.1	10.2
	no	85 90 70 95 91	86.2	9.7

Humidity = 74

$$P(\text{humidity} = 74|\text{play} = \text{yes}) = \frac{1}{\sqrt{2\pi}(10.2)} e^{-\frac{(74-79.1)^2}{2(10.2)^2}} = 0.0344$$

$$P(\text{humidity} = 74|\text{play} = \text{no}) = \frac{1}{\sqrt{2\pi}(9.7)} e^{-\frac{(74-86.2)^2}{2(9.7)^2}} = 0.0187$$

Practical Issues

- When multiplying many small fractions together you may suffer from **underflow**, resulting in the computer rounding to 0
- To account for this, it is common to take the [natural] log of probabilities and sum them: log(a*b) = log(a) + log(b)
  - Remember: all we care about is the argmax for classification

Checkup

- ML task(s)?
  - Classification: binary/multi-class?
- Feature type(s)?
- Implicit/explicit?
- Parametric?
- Online?

Summary: Naïve Bayes

- Practicality
  - Easy, generally applicable
    - May benefit from properly modeling the likelihoods
  - Very popular

- Efficiency
  - Training: relatively fast, batch
  - Testing: typically very fast
    - Assuming cached distributions [parameters]

- Performance
  - Optimal in some situations, often very good (common for use in NLP, such as spam detection)