# Lecture 15: Naïve Bayes Classifiers

Madhavan Mukund

https://www.cmi.ac.in/~madhavan

Data Mining and Machine Learning
August–December 2020

# Bayesian classifiers

- As before
  - Attributes $\{A_1, A_2, \ldots, A_k\}$ and
  - Classes $C = \{c_1, c_2, \ldots c_\ell\}$
- Each class $c_i$ defines a probabilistic model for attributes
  - $Pr(A_1 = a_1, \ldots, A_k = a_k \mid C = c_i)$
- Given a data item $d = (a_1, a_2, \ldots, a_k)$, identify the best class $c$ for $d$
- Maximize $Pr(C = c_i \mid A_1 = a_1, \ldots, A_k = a_k)$

# Generative models

- To use probabilities, need to describe how data is randomly generated
  - Generative model
- Typically, assume a random instance is created as follows
  - Choose a class $c_j$ with probability $Pr(c_j)$
  - Choose attributes $a_1, \ldots, a_k$ with probability $Pr(a_1, \ldots, a_k \mid c_j)$
- Generative model has associated parameters $\theta = (\theta_1, \ldots, \theta_m)$
  - Each class probability $Pr(c_j)$ is a parameter
  - Each conditional probability $Pr(a_1, \ldots, a_k \mid c_j)$ is a parameter
- We need to estimate these parameters

# Maximum Likelihood Estimators

- Our goal is to estimate parameters (probabilities) $\theta = (\theta_1, \ldots, \theta_m)$
- Law of large numbers allows us to estimate probabilities by counting frequencies
- Example: Tossing a biased coin, single parameter $\theta = Pr(\text{heads})$
  - $N$ coin tosses, $H$ heads and $T$ tails
  - Why is $\hat{\theta} = H/N$ the best estimate?
- Likelihood
  - Actual coin toss sequence is $\tau = t_1 t_2 \ldots t_N$
  - Given an estimate of $\theta$, compute $Pr(\tau \mid \theta)$ — likelihood $L(\theta)$
- $\hat{\theta} = H/N$ maximizes this likelihood — $\arg\max_{\theta} L(\theta) = \hat{\theta} = H/N$
  - Maximum Likelihood Estimator (MLE)

# Bayesian classification

- Maximize $Pr(C = c_i \mid A_1 = a_1, \ldots, A_k = a_k)$

- By Bayes' rule,

$$Pr(C = c_i \mid A_1 = a_1, \ldots, A_k = a_k)$$

$$= \frac{Pr(A_1 = a_1, \ldots, A_k = a_k \mid C = c_i) \cdot Pr(C = c_i)}{Pr(A_1 = a_1, \ldots, A_k = a_k)}$$

$$= \frac{Pr(A_1 = a_1, \ldots, A_k = a_k \mid C = c_i) \cdot Pr(C = c_i)}{\sum_{j=1}^{\ell} Pr(A_1 = a_1, \ldots, A_k = a_k \mid C = c_j) \cdot Pr(C = c_j)}$$

- Denominator is the same for all $c_i$, so sufficient to maximize

$$Pr(A_1 = a_1, \ldots, A_k = a_k \mid C = c_i) \cdot Pr(C = c_i)$$

# Example

- To classify $A = g, B = q$

- $Pr(C = t) = 5/10 = 1/2$
- $Pr(A = g, B = q \mid C = t) = 2/5$
- $Pr(A = g, B = q \mid C = t) \cdot Pr(C = t) = 1/5$

- $Pr(C = f) = 5/10 = 1/2$
- $Pr(A = g, B = q \mid C = f) = 1/5$
- $Pr(A = g, B = q \mid C = f) \cdot Pr(C = f) = 1/10$

- Hence, predict $C = t$

| $A$ | $B$ | $C$ |
|---|---|---|
| $m$ | $b$ | $t$ |
| $m$ | $s$ | $t$ |
| $g$ | $q$ | $t$ |
| $h$ | $s$ | $t$ |
| $g$ | $q$ | $t$ |
| $g$ | $q$ | $f$ |
| $g$ | $s$ | $f$ |
| $h$ | $b$ | $f$ |
| $h$ | $q$ | $f$ |
| $m$ | $b$ | $f$ |

# Example . . .

- What if we want to classify $A = m, B = q$?
- $Pr(A = m, B = q \mid C = t) = 0$
- Also $Pr(A = m, B = q \mid C = f) = 0$!
- To estimate joint probabilities across all combinations of attributes, we need a much larger set of training data

| $A$ | $B$ | $C$ |
|---|---|---|
| $m$ | $b$ | $t$ |
| $m$ | $s$ | $t$ |
| $g$ | $q$ | $t$ |
| $h$ | $s$ | $t$ |
| $g$ | $q$ | $t$ |
| $g$ | $q$ | $f$ |
| $g$ | $s$ | $f$ |
| $h$ | $b$ | $f$ |
| $h$ | $q$ | $f$ |
| $m$ | $b$ | $f$ |

# Naïve Bayes classifier

- Strong simplifying assumption: attributes are pairwise independent

$$Pr(A_1 = a_1, \ldots, A_k = a_k \mid C = c_i) = \prod_{j=1}^{k} Pr(A_j = a_j \mid C = c_i)$$

  - $Pr(C = c_i)$ is fraction of training data with class $c_i$
  - $Pr(A_j = a_j \mid C = c_i)$ is fraction of training data labelled $c_i$ for which $A_j = a_j$

- Final classification is

$$\arg\max_{c_i} \; Pr(C = c_i) \prod_{j=1}^{k} Pr(A_j = a_j \mid C = c_i)$$

# Naïve Bayes classifier . . .

- Conditional independence is not theoretically justified
- For instance, text classification
  - Items are documents, attributes are words (absent or present)
  - Classes are topics
  - Conditional independence says that a document is a set of words: ignores sequence of words
  - Meaning of words is clearly affected by relative position, ordering
- However, naive Bayes classifiers work well in practice, even for text classification!
  - Many spam filters are built using this model

# Example revisited

- Want to classify $A = m, B = q$
- $Pr(A = m, B = q \mid C = t) = Pr(A = m, B = q \mid C = f) = 0$
- $Pr(A = m \mid C = t) = 2/5$
- $Pr(B = q \mid C = t) = 2/5$
- $Pr(A = m \mid C = f) = 1/5$
- $Pr(B = q \mid C = f) = 2/5$
- $Pr(A = m \mid C = t) \cdot Pr(B = q \mid C = t) \cdot Pr(C = t) = 2/25$
- $Pr(A = m \mid C = f) \cdot Pr(B = q \mid C = f) \cdot Pr(C = f) = 1/25$
- Hence predict $C = t$

| $A$ | $B$ | $C$ |
|---|---|---|
| $m$ | $b$ | $t$ |
| $m$ | $s$ | $t$ |
| $g$ | $q$ | $t$ |
| $h$ | $s$ | $t$ |
| $g$ | $q$ | $t$ |
| $g$ | $q$ | $f$ |
| $g$ | $s$ | $f$ |
| $h$ | $b$ | $f$ |
| $h$ | $q$ | $f$ |
| $m$ | $b$ | $f$ |

# Zero counts

- Suppose $A = a$ never occurs in the test set with $C = c$
- Setting $Pr(A = a \mid C = c) = 0$ wipes out any product $\prod_{i=1}^{k} Pr(A_i = a_i \mid C = c)$ in which this term appears
- Assume $A_i$ takes $m_i$ values $\{a_{i1}, \ldots, a_{im_i}\}$
- "Pad" training data with one sample for each value $a_j$ — $m_i$ extra data items
- Adjust $Pr(A_i = a_i \mid C = c_j)$ to $\dfrac{n_{ij} + 1}{n_j + m_i}$

  where
  - $n_{ij}$ is number of samples with $A_i = a_i$, $C = c_j$
  - $n_j$ is number of samples with $C = c_j$

# Smoothing

- Laplace's law of succession

$$Pr(A_i = a_i \mid C = c_j) = \frac{n_{ij} + 1}{n_j + m_i}$$

- More generally, Lidstone's law of succession, or smoothing

$$Pr(A_i = a_i \mid C = c_j) = \frac{n_{ij} + \lambda}{n_j + \lambda m_i}$$

- $\lambda = 1$ is Laplace's law of succession

# Summary

- Use Bayes' Theorem to build a probabilistic classifier
- Need to define a generative model, for which frequencies are maximum likelihood estimators
- Naïve Bayes classifiers: simplifying assumption of conditional independence
  - No theoretical justification
  - Works well in practice
- Overcome zero counts using smoothing