CPSC 340 Tutorial

Week 2

September ~~13~~, 2021

21

Probability Basics I

- $p(x)$ denotes the probability of random variable X taking on the value x, short for $P(X = x)$
- $0 \leq p(x) \leq 1$
- $\sum_{x \in X} p(x) = 1$
- $p(x) = 1 - p(\neg x)$

Probability Basics II

- $p(x, y)$ denotes the joint probability of x and y both happening
  - interpreted as the intersection of the events $p(x \cap y)$
  - $p(x, y) = p(y, x)$
- $p(x \cup y) = p(x) + p(y) - p(x \cap y)$
  - Probability of at least one of them happening

Probability Basics III

- **Marginalization Rule** $p(x) = \sum_{y \in Y} p(x, y)$
  - Use it to find out $p(x)$ given its joint probability with another random variable
- **Conditional Probability** $p(y|x) = \frac{p(x,y)}{p(x)}$
  - Intuition is that knowing the condition eliminates some options
- **Product Rule** $p(x, y) = p(x|y)p(y) = p(y|x)p(x)$
  - Get this by re-arranging the conditional probability equation
- **Law of Total Probability** $p(x) = \sum_{y \in Y} p(x|y) \cdot p(y)$
  - A variant of the marginalization rule
- **Bayes Rule** $p(y|x) = \frac{p(x,y)}{p(x)} = \frac{p(x|y)p(y)}{p(x)}$

Probability Basics IV

- **Independence** $x \perp y$
  - $p(x, y) = p(x) \cdot p(y)$
  - For example, x is rolling a die and y is flipping a coin
- **Conditional Independence** $x_1 \perp x_2 | y$
  - $p(x_1, x_2 | y) = p(x_1 | y) \cdot p(x_2 | y)$
  - equivalently, $p(x_1 | x_2, y) = p(x_1 | y)$
- See Notes on probability

Naive Bayes

We want to model the probabilities of each class given a data point, then we can predict a new point's label by picking the class that give the highest posterior probability $p(y_i|x_i)$.

$$
\begin{aligned}
p(y_i|x_i) &= \frac{p(x_i|y_i)p(y_i)}{p(x_i)} \\
&\propto p(x_i|y_i)p(y_i) \\
&\approx \left[\prod_{j=1}^{d} p(x_{ij}|y_i)\right] p(y_i)
\end{aligned}
$$

where the last part comes from the naive conditional independence assumption

Naive Bayes Example

CPSC	Offer	\$	spam
0	1	0	0
0	1	1	0
1	0	0	0
1	0	0	0
1	1	0	0
0	0	1	1
0	1	1	1
1	1	1	1

- $P(\text{spam}) =$
- $P(\text{not spam}) =$
- $P(\text{CPSC} \mid \text{spam}) =$
- $P(\text{CPSC} \mid \text{not spam}) =$
- $P(\text{Offer} \mid \text{spam}) =$
- $P(\text{Offer} \mid \text{not spam}) =$
- $P(\$ \mid \text{spam}) =$
- $P(\$ \mid \text{not spam}) =$

Naive Bayes Example

- How should we classify an email with [1, 0, 1]:

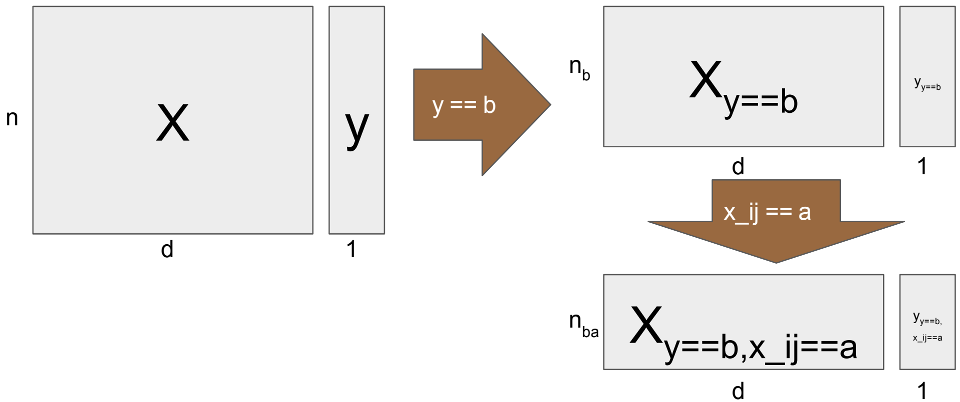
$$P(spam|CPSC, \$) = ?$$

Naive Bayes in Matrix Form

- Let each feature of X have values {1, 2, 3, …, q}
- Let y have values {1, 2, 3, …, k}

Compute:

$$P(x_{ij} = a \mid y = b)$$

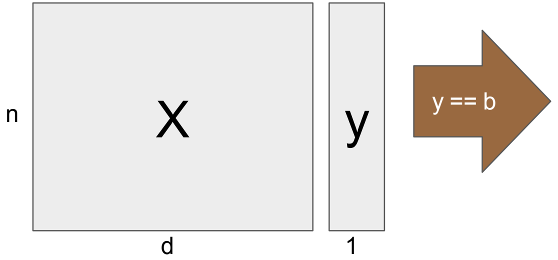


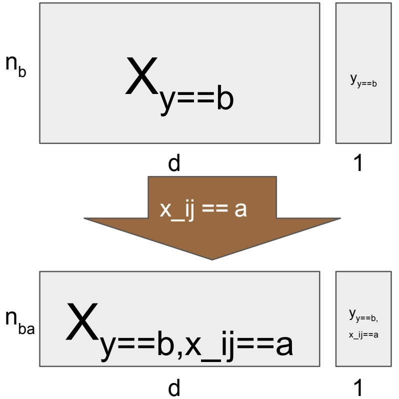
Naive Bayes in Matrix Form

- Let each feature of X have values {1, 2, 3, …, q}
- Let y have values {1, 2, 3, …, k}

Compute:

$$P(x_{ij} = a \mid y = b)$$



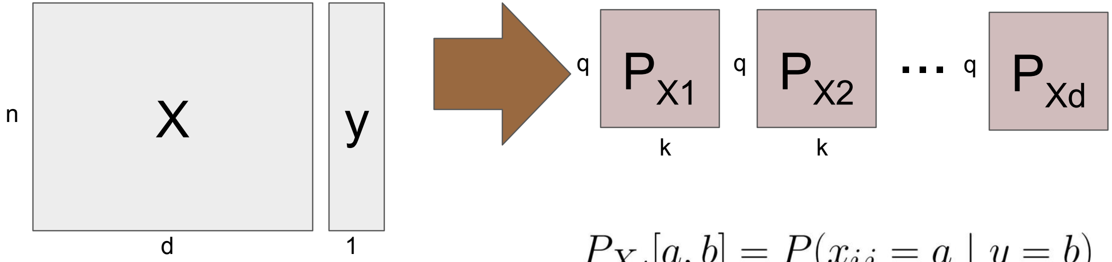


Compute:

$$P(x_{ij} = a \mid y = b)$$ = n_ba / n_b

Naive Bayes in Matrix Form

- Let each feature of X have values {1, 2, 3, …, q}
- Let y have values {1, 2, 3, …, k}



$$P_{X_j}[a, b] = P(x_{ij} = a \mid y = b)$$

Training, Testing, and Validation Set

- Given training data, we would like to learn a model to minimize error on the testing data
- How do we decide decision tree depth?
- We care about test error.
- But we can't look at test data.
- So what do we do?????

Training, Testing, and Validation Set

- Given training data, we would like to learn a model to minimize error on the testing data
- One answer: Use part of your train data to approximate test error. Split training objects into training set and validation set:
    - Train model on the training data. Test model on the validation data

Cross Validation

- Isn't it wasteful to only use part of your data? k-fold cross-validation:

  Train on k-1 folds of the data, validate on the other fold. Repeat this k times with different splits, and average the score.

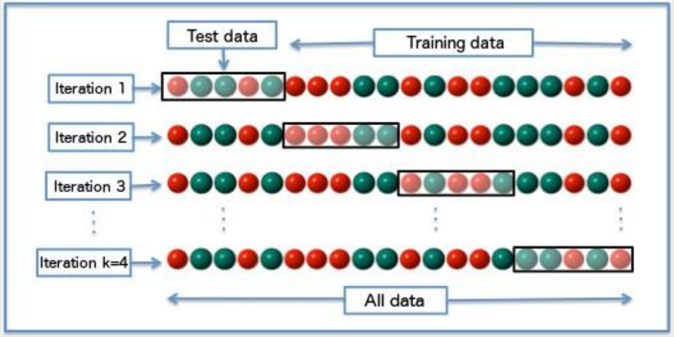


Figure 1: Adapted from Wikipedia

Note: if examples are ordered, split should be random

References

- Based off slides from 2017 and from 2021S by Nam Hee Gordon kim (and from Liconne).