Naive Bayes Classifier

Naive Bayes Introduction

Q: What classifier model works:

- when you have more features than observations?
- when you need to train and predict quickly?
- in an online setting? (i.e. continually receiving new data)

A: Naive Bayes

Outline

- Review Bayes theorem
- Review MAP estimation
- Review independence and conditional independence
- Derive Naive Bayes classifier
- Apply Naive Bayes to document classification
- Understand nuances of Naive Bayes

Bayes Theorem Review

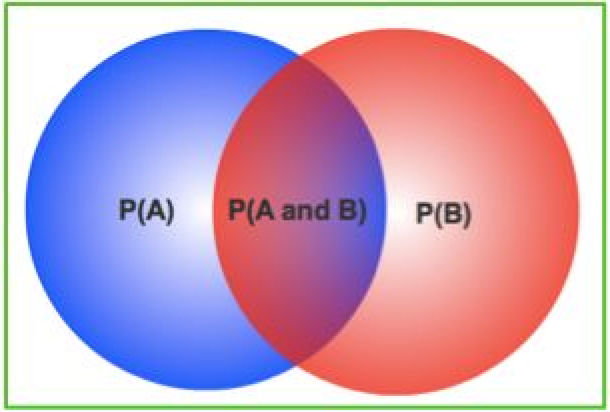
Problem Motivation

- How to relate conditional probabilities between two events?
  - What's the relationship between $P(A \mid B)$ and $P(B \mid A)$?
- How to incorporate prior knowledge and belief into interpretation of data?

→ Use Bayes Theorem

Conditional Probability Review

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$



Bayes Theorem Derivation

Definition of conditional probability: $P(A|B) = \dfrac{P(A \cap B)}{P(B)}$

Property of joint probability: $P(B|A)P(A) = P(A \cap B)$

→ Bayes Theorem: $P(A|B) = \dfrac{P(B|A)P(A)}{P(B)}$

Bayes Theorem Explanation

$$\underset{\text{Posterior distribution}}{\boxed{P(A|B)}} = \frac{\overset{\text{Likelihood}}{\boxed{P(B|A)}}\overset{\text{Prior}}{\boxed{P(A)}}}{P(B)}$$

Example: Relating Prior Belief to Data

You have a drawer of 100 coins, 10 of which are biased.

P(heads | fair coin) = .5

P(heads | biased coin) = .25

You randomly choose a coin and flip it three times. It comes up heads all three times.

What is P(fair coin | H, H, H)?

Example: Relating Prior Belief to Data

$$\overset{\text{Posterior}}{\downarrow}\boxed{P(\text{fair coin}|HHH)} = \frac{\overset{\text{Likelihood}}{\downarrow}\boxed{P(HHH|\text{fair coin})}\ \overset{\text{Prior}}{\downarrow}\boxed{P(\text{fair coin})}}{P(HHH)}$$

Example: Relating Prior Belief to Data

$$\text{Prior} = P(\text{fair coin}) = \frac{\#\text{ of fair coins}}{\#\text{ of all coins}} = 90\%$$

$$\text{Likelihood} = P(HHH|\text{fair coin}) = \left(\frac{1}{2}\right)^3$$

Example: Relating Prior Belief to Data

$$P(\text{fair coin}|HHH) = \frac{P(HHH|\text{fair coin})P(\text{fair coin})}{\boxed{P(HHH)}}$$

Calculated from Law of Total Probability

Example: Relating Prior Belief to Data

Law of Total Probability:

$$P(Y) = P(Y|X)P(X) + P(Y|X^c)P(X^c)$$

$$P(HHH) = P(HHH|\text{fair coin})P(\text{fair coin}) + P(HHH|\text{unfair coin})P(\text{unfair coin})$$

$$= .5^3 * .9 + .25^3 * .1$$

$$\approx .114$$

MAP Estimation Review

Maximum A Posteriori (MAP)

Recall Bayes Rule:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

MAP finds H to maximize P(H | X):

$$\underset{H}{\operatorname{argmax}}\ P(H|X) = \underset{H}{\operatorname{argmax}}\ \frac{P(X|H)P(H)}{P(X)}$$

$$= \underset{H}{\operatorname{argmax}}\ P(X|H)P(H)$$

Relating Prior Knowledge/Belief to Data

You have a drawer of 100 coins, 10 of which are biased.

P(heads | fair coin) = .5

P(heads | biased coin) = .25

You randomly choose a coin and flip it once. It comes up heads three times.

Which coin type (fair or unfair) is most probable under the posterior?

Example: Relating Prior Belief to Data

$$P(\text{fair coin}|HHH) = \frac{.5^3 * .9}{.114} = .987$$

$$P(\text{unfair coin}|HHH) = 1 - \frac{.5^3 * .9}{.114} = .013$$

Maximum A Posteriori (MAP)

Recall Bayes Rule:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

MAP finds H to maximize P(H | X):

$$\operatorname*{argmax}_H P(H|X) = \operatorname*{argmax}_H \frac{P(X|H)P(H)}{P(X)}$$

$$= \operatorname*{argmax}_H P(X|H)P(H)$$

Independence and Conditional Independence Review

Independence

$$P(A \cap B) = P(A)P(B)$$

Conditional Independence

$$P(A \cap B|C) = P(A|C)P(B|C)$$

Generative vs Discriminative Models

Discriminative Model

- Only estimates conditional distribution

$$P(Y|X)$$

Generative Models

- Estimates the joint distribution
- Can generate new (synthetic) data by sampling from joint distribution

$$P(Y|X)P(X) = P(Y \cap X)$$

Naive Bayes Classifier

Derivation

Bayes Rule: $P(Y|X) = \dfrac{P(X|Y)P(Y)}{P(X)}$

MAP Estimation: $\underset{Y}{\operatorname{argmax}}\ P(Y|\vec{X}) = \underset{Y}{\operatorname{argmax}}\ \dfrac{P(\vec{X}|Y)P(Y)}{P(\vec{X})}$

$$= \underset{Y}{\operatorname{argmax}}\ P(\vec{X}|Y)P(Y)$$

Conditional Independence: $P(\vec{X}|Y) = P(x_1|Y)P(x_2|Y)...P(x_p|Y)$

$$\boxed{\text{Naive Bayes Classifier: } \underset{Y}{\operatorname{argmax}}\ P(Y|\vec{X}) = \underset{Y}{\operatorname{argmax}}\ P(x_1|Y)P(x_2|Y)...P(x_3|Y)P(Y)}$$

Summary

$$\text{Naive Bayes Classifier: } \underset{Y}{\operatorname{argmax}}\ P(Y|\vec{X}) = \underset{Y}{\operatorname{argmax}}\ P(x_1|Y)P(x_2|Y)...P(x_3|Y)P(Y)$$
$$= \underset{Y}{\operatorname{argmax}}\ P(Y)\prod_{i=1}^{k} P(x_i|Y)$$

Naive Bayes Classifier is MAP estimation combined with conditional independence

Document Classification with Naive Bayes

Problem Motivation

How to predict what topic a given document is about?

Example Document:

> *"The Giants won the World Series."*

Q: How can we decide whether this document is fiction or nonfiction?
A: Use word counts from corpus of labeled fiction or nonfiction documents to train Naive Bayes model.

Multinomial Event Model

- Author randomly picks a category (e.g. fiction, nonfiction) according to prior distribution **P(Y)**
- Then randomly draws from bag of words with replacement according conditional distribution **P(X|Y)**

Estimating Prior Distribution

- Prior is discrete distribution over all classes
- Use sample (corpus) proportion to estimate prior

$$P(y = "fiction") = \frac{\text{number of fiction documents}}{\text{total number of documents}}$$

Estimating Conditional Distribution

Fiction Corpus:

"the cat in the hat"

"the cat in the tree"

"the cow jumped over the moon"

P(word = "cat" | fiction) = 2/16

P(word = "jumped" | fiction) = 1/16

Estimating Conditional Distribution

Nonfiction Corpus:

"the giants won the game"

"the stock market was up today"

"the candidate won the election"

P(word = "giants" | nonfiction) = 1/16

P(word = "won" | nonfiction) = 2/16

Example: “The Cat in the Hat”

$$\underset{Y}{\operatorname{argmax}}\; P(y|doc = \text{"the cat in the hat"}) =$$

$$= \underset{Y}{\operatorname{argmax}}\; P(doc = \text{"the cat in the hat"}|y)P(y)$$

Example: “The Cat in the Hat”

$$= \underset{Y}{\text{argmax}}\ P(doc = \text{"the cat in the hat"}|y)P(y)$$

$$= \underset{Y}{\text{argmax}}\ P(y)P(\text{"}the\text{"}|y)P(\text{"}cat\text{"}|y)P(\text{"}in\text{"}|y)P(\text{"}the\text{"}|y)P(\text{"}hat\text{"}|y)$$

$$= \underset{Y}{\text{argmax}}\ P(y)P(\text{"}the\text{"}|y)^2P(\text{"}cat\text{"}|y)^1P(\text{"}in\text{"}|y)^1P(\text{"}hat\text{"}|y)^1$$

$$= \underset{Y}{\text{argmax}}\ P(y)\prod_{w \in vocab} P(w|y)^{x_w} =$$

Naive Bayes Details

Naive Bayes Details

- Log-transformation
- Dealing with unknown words
- Laplace smoothing
- Online learning
- Extensions
- When to use Naive Bayes

Log-Transformation

- Very small number:

$$P(y) \prod_{w \in vocab} P(w|y)^{x_w}$$

- Risk of numerical underflow
- Use log probabilities instead:

$$log(P(y)) + \sum_{w \in vocab} x_w log(P(w|y))$$

Laplace (add 1) Smoothing

$$P(y) \prod_{w \in vocab} P(w|y)^{x_w}$$

Q: What happens if a word from a new document doesn't appear in a class in the training corpus?

A: P(word | class) = 0 $\rightarrow$ estimated P(class | word) = 0

Laplace (add 1) Smoothing

- Add 1 to each word's frequency
- As if we saw each word more than we actually did

$$P(x|c) = \frac{(\#\text{ of times } x \text{ appears in docs of class c}) + 1}{(\text{total } \#\text{ of words in docs of class c}) + (\text{total } \#\text{ words in vocabulary})}$$

Unknown Words

- Add generic [unknown word] to the vocabulary
- Gives small positive likelihood to any word not previously seen

$$P(x_{unknown}|c) = \frac{(\text{\# of times } x_{unknown} \text{ appears in docs of class c}) + 1}{(\text{total \# of words in docs of class c}) + (\text{total \# words in vocabulary+1})}$$

$$= \frac{1}{(\text{total \# of words in docs of class c}) + (\text{total \# words in vocabulary+1})}$$

Online Learning

- What happens when new documents are added to the corpus?
- Just increment the word counts

Old doc: "the giants won the game"
Old doc: "the stock market was up today"
New doc: "the candidate won the election"

Old: P(word = "won" | nonfiction) = 1/11
New: P(word="won" | nonfiction) = (1 + 1) / (11 + 5) = 2/16

Extensions

- Can use other conditional distributions (Gaussian, etc.)
- Can use feature weighting

Details: "Tackling the Poor Assumptions of Naive Bayes Classifiers" http://machinelearning.wustl.edu/mlpapers/paper_files/icml2003_RennieSTK03.pdf

When to use Naive Bayes?

Pros

- Good with “wide data”
  (i.e. more features than observations)
- Fast to train / good at online learning
- Simple to implement

Cons

- Can be hampered by correlated features
- Probabilistic estimates are unreliable because of naive assumption
- Sometimes outperformed by other models