Naive Bayes Classifier

Bayes Classifier

Assumes we know the true distribution of the data

Assign observations to the class that is most probable

↳ For observation data x_0, we assign it to class K if:

$$\mathbb{P}(Y=k \mid X=x_0) \geq \mathbb{P}(Y=l \mid X=x_0) \text{ for every class } l$$

We use Bayes Rule to calculate $\mathbb{P}(Y=k \mid X=x_0)$

↳ Let $\pi_l = \mathbb{P}(Y=l)$ be the prior probability that a randomly chosen observation comes from class l

Let $f_l(x) = \mathbb{P}(X=x \mid Y=l)$ be the density of X for an observation that comes from class l (likelihood)

$$\Rightarrow \mathbb{P}(X=x) = \sum_{l=1}^{K} \mathbb{P}(X=x, Y=l) = \sum_{l=1}^{K} \mathbb{P}(X=x \mid Y=l)\,\mathbb{P}(Y=l)$$

$$= \sum_{l=1}^{K} f_l(x)\,\pi_l$$

Want: $$\mathbb{P}(Y=k \mid X=x) = \frac{\mathbb{P}(X=x \mid Y=k)\,\mathbb{P}(Y=k)}{\mathbb{P}(X=x)}$$

$$= \frac{f_k(x)\,\pi_k}{\sum_{l=1}^{K} f_l(x)\,\pi_l}$$

is the posterior probability that observation x belongs to class k

Note: π_k can be estimated from our training data

But f_k often cannot, especially if there are a large number of features

Even though the Bayes Classifier is the best possible classifier, it is (almost always) entirely theoretical when attempted on real data

Naive Bayes Classifier

Idea: Add an independence assumption to the Bayes Classifier so that the likelihood $f_K(x)$ can be estimated from data

$f_K(x) = \mathbb{P}(X = x \mid Y = K)$ is "hard" because we may not have enough data to consider all input combinations for estimation

"Naive" assumption: all predictors are independent (conditional on the class being known)

$$\hookrightarrow f_K(x) = \mathbb{P}(X = x \mid Y = K)$$

$$= \underbrace{\mathbb{P}(X_1 = x_1 \mid Y = K)}_{f_{K1}(x_1)} \underbrace{\mathbb{P}(X_2 = x_2 \mid Y = K)}_{f_{K2}(x_2)} \cdots \underbrace{\mathbb{P}(X_p = x_p \mid Y = K)}_{f_{Kp}(x_p)}$$

with p features

$f_{ki}(x_i) = \mathbb{P}(X_i = x_i \mid Y = k)$ is "easy" to estimate through counting or Kernel estimation [on training data]

$\Rightarrow$ Estimate the posterior probability $\mathbb{P}(Y = k \mid X = x)$ by:

$$\frac{\left(\prod_{i=1}^{p} f_{ki}(x_i)\right)\pi_k}{\sum_{l=1}^{K}\left(\prod_{i=1}^{p} f_{li}(x_i)\right)\pi_l}$$

Often ignore the denominator since it is constant for a given x. Additionally, can be considered as the log-probability:

$\hookrightarrow k^* = \underset{k}{\operatorname{argmax}} \left[\log(\pi_k) + \sum_{i=1}^{p} \log(f_{ki}(x_i))\right]$

with predicted class as the one with the greatest posterior probability

In Text Classification

ex: Term Frequency: $tf(t, d)$ = count of term t in document d

(sometimes normalize by # terms in the document d)

$$\hookrightarrow \hat{f}_{k_i}(x_i) = \frac{\sum_{d \in [k]} tf(x_i, d) + \alpha}{\sum_{d \in [k]} N_d + \alpha V}$$

Smoothing

- α is an additive smoother ($\alpha = 1$ for Laplace)
- V is the size of the vocabulary in the training data

where:
- x_i is a token from the feature vector x
- N_d is the total number of terms in document d
- $[k]$ is the set of documents in class k

Text Mining

Text mining is a process through which users derive information from a given piece of text

Text Mining Methodologies

1) Sentiment Analysis (Discussed briefly last week + in more depth next week)

Idea: Extract the underlying opinion within textual data

Sometimes referred to as opinion mining

Often focused on polarity detection to determine, e.g., positive/negative

2 Main Approaches

a) Manual Tagging / Lexicon-Based / Dictionary-Based

b) Automatic Tagging / Machine-Learning-Based

↳ often used with Naive Bayes, Support Vector Machines, or Random Forest

2) Information Extraction

Idea: Extract predefined data types from text documents

Example: Named-Entity Recognition to match documents to companies

Often applied to official documents (ex. 10-K's) to recover the relevant information

3) Topic Modeling (Will discuss more in 2 weeks)

Idea: Identify "topics" that best describe a set of documents

Typically unsupervised classification problem

↳ similar to clustering numerical data

Provides a method for organizing + summarizing large collections of documents

Applications in Finance

Overview

- Financial Prediction: Use textual data as features to improve
  market prediction

- Banking: Use text data as features in
  a) Fraud/Money laundering detection
  b) Customer Relationship Management
  c) Risk management

- Corporate Finance: Use text data as features in:
  a) Fraud detection
  b) Sustainability analysis
  c) Review of reports

Financial Prediction

Many studies have evaluated the use of textual data as additional features for predicting market movements

Typically utilizes sentiment analysis for quantifying textual data

General Outcomes:

- Model performance is better when using both text + traditional data than either individually
- Better results found with smaller / less liquid markets
  - ↪ Think about meme stocks as well
- Use official reports to get frequent estimates of (infrequent) macroeconomic statistics

Difficulty: Appropriate training of the sentiment analyzer

↪ create dictionary or training data

Banking

Multitude of applications within traditional banking

↳ Detecting money laundering + Know your customer (KYC) rules

↘ extract profiles of entities that could be suspect

↘ does not make final decisions

↳ Loan decisions + credit scores

↘ estimate, e.g., the financial obligations for different applicants

↘ review social media posts for "risky" signals

↘ sometimes supervised, sometimes unsupervised

Corporate Finance

Text mining corporate reports + earnings calls can provide important information

↳ algorithms can "read" faster than humans can to extract relevant features

↳ sensitive to slight changes in wording [can be good or bad]

↳ evidence that some companies are running reports through sentiment analyzers before being released to make sure it is consistent with the intent

[Sometimes the best metrics are really simple:
if format changes, then it is possible that the company wants to hide bad information]

Also used for corporate fraud detection, etc.

Primary Challenges

- Restrictions on confidential data
- Absence of well-defined financial lexicon / dictionary (& can be company specific)
- Infrequent release of official reports can lead to overfitting
- Can include lots of redundant data that appears independent (e.g., on social media)
- (If social media) sarcasm & vernacular are difficult to parse