Applied Machine Learning

Classification - Naive Bayes

Naive Bayes

- Bayesian Classification
- How to estimate the probability of a class from a data sample
- How to incorporate probability distributions in the estimation
- How to choose models and parameters

Bayes Classification

- Calculates the probability of a class
    - Combination of prior knowledge with observed data
    - Prediction of multiple classes can be a weighted combination of each
- Each training example can modify the probability of a class

Bayes Classification

- Each class label y has a probability distribution
- Test example to classify: $X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(N)} \end{bmatrix}$
- Report the label y with the highest probability $y = argmax_{y \in Y} P(y \mid X)$

Bayes Classification

- Cost is linear in the number of classes y
- Constructing $P(y \mid X)$
  - can be learned from available data
  - consider underlying distribution

When to use a Naive Bayes Classifier

- Very good technique to make probabilistic predictions
  - 90% chance of a picture corresponding to some digit
  - 70% chance that a patient has some disease based on lab test results
- Effective with high dimensional data
- It is robust to incomplete data
- It is easy to implement
- Competitive against other techniques

Naive Bayes

- Finding the most probable class: $y = argmax_{y \in Y} P(y \mid X)$
- Bayes Theorem: $P(y \mid X) = \dfrac{P(X \mid y)P(y)}{P(X)}$
- Finding the most probable class: $y = argmax_{y \in Y} \dfrac{P(X \mid y)P(y)}{P(X)}$

$$y = argmax_{y \in Y} \frac{P(X|y)P(y)}{P(X)}$$

$$P(X|y)$$

- X is a lab test composed of features $x^{(i)}$
  - Example: each $x^{(i)}$ is a measurement in the lab test
- Naive assumption
  - Features $x^{(i)}$ are independent of each other conditioned on class y
  - Example: estimate probability of each measurement conditioned on being sick

$$X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(N)} \end{bmatrix} = \begin{bmatrix} 0.5 & 100 & \dots & 14 \\ 0.9 & 90 & \dots & 12 \\ \vdots & \vdots & \ddots & \vdots \\ 0.6 & 121 & \dots & 15 \end{bmatrix}$$

$$P(X|y) \quad = P(x^{(1)}|y)P(x^{(2)}|y)\dots P(x^{(N)}|y)$$

$$P(X|y) \quad = \prod_i P(x^{(i)}|y)$$

$$y = argmax_{y \in Y} \frac{[\prod_i P(x^{(i)}|y)]P(y)}{P(X)}$$

$$y = argmax_{y \in Y} \frac{P(X|y)P(y)}{P(X)}$$

$$P(y)$$

- $P(y)$ from the distribution of classes
    - Example: frequency of tests that correspond to being sick

$$y = argmax_{y \in Y} \frac{[\prod_i P(x^{(i)}|y)]P(y)}{P(X)}$$

$$y = argmax_{y \in Y} \frac{P(X \mid y)P(y)}{P(X)}$$

$$P(X)$$

- $P(X)$ is probability of observing data point X
  - Hard to obtain
  - Example: probability of specific lab test results
- Independent on the class label y
  - Not needed for finding the class with maximum probability

$$y = argmax_{y \in Y}[\prod_i P(x^{(i)} \mid y)]P(y)$$

$$y = argmax_{y \in Y}[\prod_i P(x^{(i)} | y)]P(y)$$

Numerical Issues

- Products of probabilities may become too small
- Transform products into sums through logarithms
  - Preserve relative differences among classes
- $P(x^{(i)} | y)$
  - fit $x^{(i)}$ in probability distribution
- $P(y)$
  - fit y in probability distribution

$$y = argmax_{y \in Y}[\prod_i P(x^{(i)} | y)]P(y)$$

$$y = argmax_{y \in Y} \log[[\prod_i P(x^{(i)} | y)]P(y)]$$

$$y = argmax_{y \in Y} \sum_i [\log P(x^{(i)} | y)] + \log[P(y)]$$

Fitting data in Probability Distributions

- 0-1: Bernoulli
- Count: Poisson
- Discrete: Multinomial model
- Real Valued
  - Normal distribution
    - even If dataset does not look normal but the normals split classes
  - Quantized Multinomial model
    - may be better if there is significant overlap

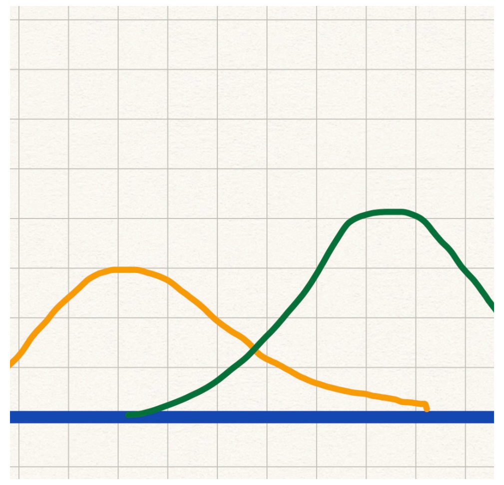
Fitting data in Probability Distributions

- 0-1: Bernoulli
- Count: Poisson
- Discrete: Multinomial model
- Real Valued
  - Normal distribution
    - even If dataset does not look normal but the normals split classes
  - Quantized Multinomial model
    - may be better if there is significant overlap

Potential Case: Fit $P(x^{(i)} | y)$ in Gaussian Distribution

- Parameters
  - $x^{(i)}$ conditioned on class y: (μ_y, σ_y)
    - Example: compute mean and standard deviation of each measurement for test lab results labeled as sick
  - From $x^{(i)}$ with class label y in the training set
- Gaussian distribution:

$$P(x^{(i)} | \mu_y, \sigma_y) = \frac{1}{\sigma_y\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(x^{(i)} - \mu_y)^2}{\sigma_y^2}}$$



$$y = argmax_{y \in Y} \sum_i [\log P(x^{(i)} | y)] + \log[P(y)]$$

$$y = argmax_{y \in Y} \sum_i [\log[\frac{1}{\sigma_y\sqrt{2\pi}}] - \frac{1}{2}\frac{(x^{(i)} - \mu_y)^2}{\sigma_y^2}] + \log[P(y)]$$

Naive Bayes Classifier

$$y = argmax_{y \in Y} + \log[\frac{N}{\sigma_y\sqrt{2\pi}}] - \sum_i [\frac{1}{2}\frac{(x^{(i)} - \mu_y)^2}{\sigma_y^2}] + \log[P(y)]$$

Incomplete Records in Train Set

- Records in the Train Set may be incomplete
  - Maybe some feature was not collected
- When fitting $P(x^{(i)} | y)$ or $P(y)$
  - Options
    - Ignoring the whole record reduces the size of the Train Set
    - Replacing with a particular value may skew the distribution
    - Best option: Ignoring the missing value affects all classes the same way

$$X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(N)} \end{bmatrix} = \begin{bmatrix} 0.5 & 100 & ? \\ 0.9 & ? & 12 \\ 0.8 & 110 & 13.5 \\ ? & 121 & 15 \end{bmatrix}$$

Models: Selection Among Choices

- There may be several options on the Model to fit $P(x^{(i)} \mid y)$
  - Distribution
  - Parameters for the distribution
- Compute cross-validated error for each model and choose the best one

Cross-Validation

```
|  Cross Validation Train  |  Validation  | Test |
|  Validation  |                          |  Test  |
|             |  Validation  |            | Test  |
|                          |  Validation |  Test  |
```

Each model is evaluated as follows

- Split data into two parts:
  - Test Set for estimation of future performance
  - Train Set for cross-validation. This set will iteratively split into folds
    - For each candidate model, iteratively
      - generate a new Fold from Train Set with a Cross-Validation Train Set and Validation Set
      - obtain parameters for model of choice with the Cross-Validation Train Set
      - evaluate with the Validation Set and record error for current Fold
    - Cross-Validation Error for chosen model is average error over all the Folds

Model Selection

- Select model that has the “best” cross-validation error
  - low error
  - error variance
- Recompute parameters of selected model with whole Train Set
  - it was split for cross-validation
- Estimate future performance with original Test Set

Naive Bayes

- Bayesian Classification
- How to estimate the probability of a class from a data sample
- How to incorporate probability distributions in the estimation
- How to choose models and parameters

Applied Machine Learning

Classification - Naive Bayes