Midterm 2 Review

- PCA
- Naive Bayes
- Logistic regression and cross entropy

Study guide is posted,
We talked about data visualization to effectively sha

PCA

- Transforms p-dimensional data to n dimensions, dimensionality reduction
- PC1 explains the most variance, PC
- PCA is a linear transformation
- Dimensionality of the transformed data depends on the number of eigenvectors we find, usually just used the first two columns
- PCA is helpful for data visualization and inferring qualitative relationships between groups
- PCs don't carry information mostly used qualitatively to see how close different data points are

Naive Bayes

- Bayes theorem P(A,B) = P(A | B) P(B), P(A,B) = P(B | A) P(A)
  P(A,B), joint probability
  P(A|B), conditional probability
  P(A), Marginal probability

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

  P(A|B), posterior
  P(B|A), likelihood
  P(A), prior
  P(B), evidence

  Independence: P(A)P(B) = P(A,B)

  Conditional Independence P(A | B,C) = P(A|C)
  Naive Bayes is naive bayes since it assumes conditional independence

  Goal of Naive bayes:
  Predict the label given a vector of features
  P(x1,x2,x3 | y) = P(x1|y) P(x2,x3|x1,y) = … = P(x1|y)P(x2|y)P(x3|y)

Naïve Bayes Model

$$p(y = k|\boldsymbol{x}) \propto p(y = k) \prod_{j=1}^{p} p(x_j|y = k).$$

Naïve Bayes Prediction

$$\hat{y} = \underset{k \in \{1,2,\cdots,K\}}{\arg\max} \; p(y = k) \prod_{j=1}^{p} p(x_j|y = k).$$

Laplace counts, used to avoid 0 probabilities

Estimating prior: p(y=k)

$$\theta_k = \frac{N_k + 1}{n + K}$$

Estimating likelihood: $p(x_j=v \mid y=k)$

$$\theta_{k,j,v} = \frac{N_{k,j,v} + 1}{N_k + |f_j|}$$

Logistic Regression

3 pieces for SGD

- Hypothesis function

  Sigmoid function 1/(1+e^-1)

  Transforms continuous number to range between 0,1 which can be used for binary classification

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = p(y = 1|\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{w}\cdot\boldsymbol{x}}}$$

  Output of the hypothesis function is the probability of the label being 1

- Cost function

  Binary cross entropy function, tells us how good our model is at predicting label. It's based on the likelihood function, generally likelihood is normally maximized since cost should be minimized we multiply by -1

$$J(\boldsymbol{w}) = -\sum_{i=1}^{n} y_i \log h_{\boldsymbol{w}}(\boldsymbol{x_i}) + (1 - y_i) \log(1 - h_{\boldsymbol{w}}(\boldsymbol{x_i}))$$

  To minimize the cost we use the derivative at a single data then adjust the weight vector accordingly.

$$\nabla J_{\boldsymbol{x}_i}(\boldsymbol{w}) = (h_{\boldsymbol{w}}(\boldsymbol{x_i}) - y_i)\boldsymbol{x_i}$$

  SGD pseudo code :

set $\vec{w} = \vec{0}$

while cost $J(\vec{w})$ is still changing:

  shuffle data points

  for i = 1,…,n:

    $\vec{w} \leftarrow \vec{w} - \alpha \nabla J_{\vec{x_i}}(\vec{w})$

store $J(\vec{w})$

derivative of $J(\vec{w})$ wrt x_i

Linear Regression

X is continuous, y is continuous

Polynomial Regression

X is continuous, y is continuous

Decision trees/stumps

X can be discrete or continuous values mapped to discrete values, y is discrete

ROC curve as an evaluation metric

X is discrete, y is continuous

Naive Bayes

X is discrete, y is discrete

Logistic regression

X is continuous, y is discrete

Entropy and Information Gain

X is discrete , Y is continuous

PCA

X is continuous, Y is continuous