RECITATION 7: UNSUPERVISED LEARNING NAÏVE BAYES

10-301/10-601 Introduction to Machine Learning (Summer 2023)
http://www.cs.cmu.edu/~hchai2/courses/10601
Released: July 13th, 2023
Quiz Date: July 18th, 2023
TAs: Alex, Andrew, Sofia, Tara, and Neural the Narwhal

1 Naive Bayes

By applying Bayes' rule, we can model the probability distribution $P(Y|X)$ by estimating $P(X|Y)$ and $P(Y)$.

$$P(Y|X) \propto P(Y)P(X|Y)$$

The Naive Bayes assumption greatly simplifies estimation of $P(X|Y)$ - we assume the features X_d are independent given the label. With math:

$P(X|Y) =$ ______________________________

Different Naive Bayes classifiers are used depending on the type of features.

- Binary Features: Bernoulli Naive Bayes - $X_d \,|\, Y = y \sim$ Bernoulli$(\theta_{d,y})$
- Discrete Features: Multinomial Naive Bayes - $X_d \,|\, Y = y \sim$ Multinomial$(\theta_{d,1,y}, \ldots, \theta_{d,K-1,y})$
- Continuous Features: Gaussian Naive Bayes - $X_d \,|\, Y = y \sim \mathcal{N}(\mu_{d,y}, \sigma^2_{d,y})$

We'll walk through the process of learning a Bernoulli Naive Bayes classifier. Consider the dataset below. You are looking to buy a car; the label is 1 if you are interested in the car and 0 if you aren't. There are three features: whether the car is red (your favorite color), whether the car is affordable, and whether the car is fuel-efficient.

Interested?	Red?	Affordable?	Fuel-Efficient?
1	1	1	1
0	0	1	0
0	0	1	1
1	0	0	0
0	0	1	1
0	0	1	1
1	1	1	1
1	1	0	1
0	0	0	0

1. How many parameters do we need to learn?

2. Estimate the parameters via MLE.

3. If I see a car that is red, not affordable, and fuel-efficient, would the classifier predict that I would be interested in it?

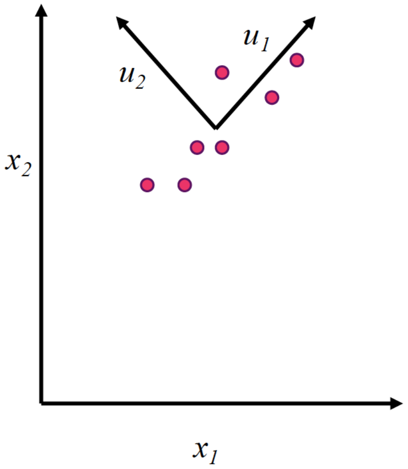
4. Is there a problem with this classifier based on your calculations for the previous question? If so, how can we fix it?

5. Now we will derive the decision boundary of a 2D Gaussian Naïve Bayes. Show that this decision boundary is quadratic. That is, show that $p(y = 1 \mid x_1, x_2) = p(y = 0 \mid x_1, x_2)$ can be written as a polynomial function of x_1 and x_2 where the degree of each variable is at most 2. You may fold *unimportant* constants into terms such as C, C', C'', C''' so long as *you are clearly showing each step*.

2 Principal Component Analysis

Principal Component Analysis aims to project data into a lower dimension, while preserving as much as information as possible.

How do we do this? By finding an orthogonal basis (a new coordinate system) of the data, then pruning the "less important" dimensions such that the remaining dimensions minimize the squared error in reconstructing the original data.



In low dimensions, finding the principal components can be done visually as seen above, but in higher dimensions we need to approach the problem mathematically. We find orthogonal unit vectors $\mathbf{v}_1 \dots \mathbf{v}_M$ such that the reconstruction error $\frac{1}{N}\sum_{i=1}^{N} ||\mathbf{x}^{(i)} - \hat{\mathbf{x}}^{(i)}||^2$ is minimized, where $\hat{\mathbf{x}}^{(i)} = \sum_{m=1}^{M}(\mathbf{v}_m^T \mathbf{x}^{(i)})\mathbf{v}_m$ are the reconstructed vectors.

If we have M new vectors and d original vectors, with $M = d$, we can reconstruct the original data with 0 error. If $M < d$, it is usually not possible to reconstruct the original data without losing any error. In other words, all the reconstruction error comes from the $M - d$ missing components. This error can be expressed in terms of the covariance matrix of the original data, and is minimized when the principal component vectors $\mathbf{v}_1 \dots \mathbf{v}_M$ are the top M eigenvectors of the covariance matrix (in terms of eigenvalues). The higher the eigenvalues for these eigenvectors are, the more information they store and the lower the reconstruction error.

For the following questions, use this Colab notebook.

Let's assume we've performed PCA on the following dataset:

Row	X1	X2	X3	X4
1	-0.21	-0.61	-0.35	0.08
2	0.15	-0.77	1.26	1.57
3	0.03	0.12	-0.39	-0.25
4	0.92	1.31	0.31	1.19
5	2.51	1.99	1.86	2.57
6	0.91	1.23	-0.01	0.04

And we've obtained the following principal components:

PC1	**PC2**	**PC3**	**PC4**
-0.53	0.23	0.48	-0.66
-0.49	0.7	-0.27	0.44
-0.43	-0.46	0.52	0.57
-0.54	-0.49	-0.65	-0.21

Which correspond to the following eigenvalues:

[3.265, 0.999, 0.043, 0.014]

1. Why are there only 4 principal components?
2. How much of the variance in the data is preserved by the first two principal components?
3. How much of the variance in the data is preserved by the first and third principal components?
4. Perform a dimensionality reduction on the points such that we project them onto the first two principal components. Then, inverse transform it back to four dimensions. What is the reconstruction error for this sample?
5. Perform a dimensionality reduction such that we project the points onto the first and third principal components. Then, inverse transform it back to four dimensions. What is the reconstruction error of this new dataset?
6. Consider the reconstruction error of the fourth row in particular. Is it lower using the first and second principal components or using the first and third? Why might this be the case?

3 K-Means

Clustering is an example of unsupervised machine learning algorithm because it serves to partition **unlabeled** data. There are many different types of clustering algorithms, but the one that is used most frequently and was introduced in class is **K-Means**.

In K-Means, we aim to minimize the objective function:

$$\sum_{i=1}^{n} \min_{j \in \{1,\ldots,k\}} ||\mathbf{x}^{(i)} - \boldsymbol{\mu}_j||^2 \tag{1}$$

Below is the K-Means algorithm:

Let $\mathcal{D} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}$ where $\mathbf{x}^{(i)} \in \mathbb{R}^d$ be the set of input examples that each have d features.

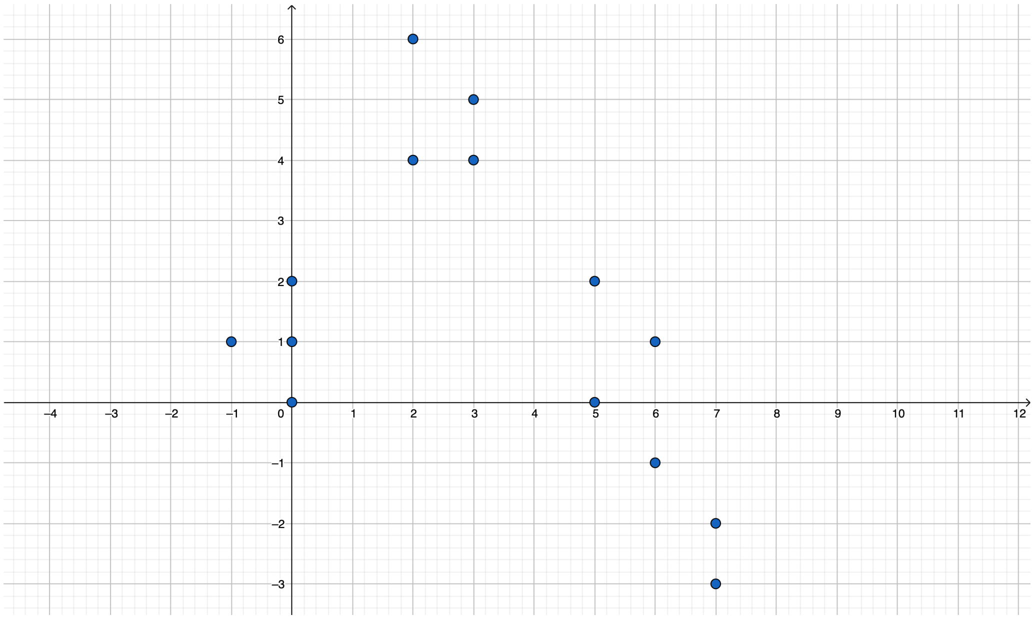
Initialize k cluster centers $\{\boldsymbol{\mu}^{(1)}, \ldots, \boldsymbol{\mu}^{(k)}\}$ where $\boldsymbol{\mu}^{(i)} \in \mathbb{R}^d$

Repeat until convergence:

1. Assign each point $\mathbf{x}^{(i)}$ to a cluster $\mathcal{C}^{(j)}$ where $j = \text{argmin}_{1 \le r \le k} ||\mathbf{x}^{(i)} - \boldsymbol{\mu}^{(r)}||$
2. Recompute each $\boldsymbol{\mu}^{(i)}$ as the mean of points in $\mathcal{C}^{(i)}$

3.1 Walking through an example

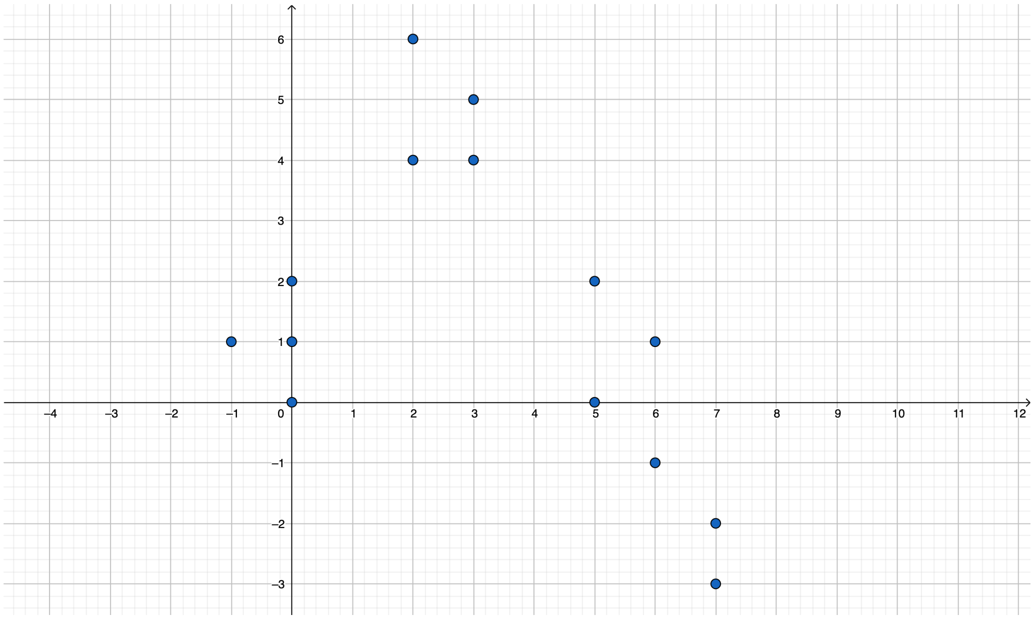
Lets walk through an example of K-Means with $k = 3$ using the following dataset for the first iteration:



Perform one iteration of the K-Means algorithm:

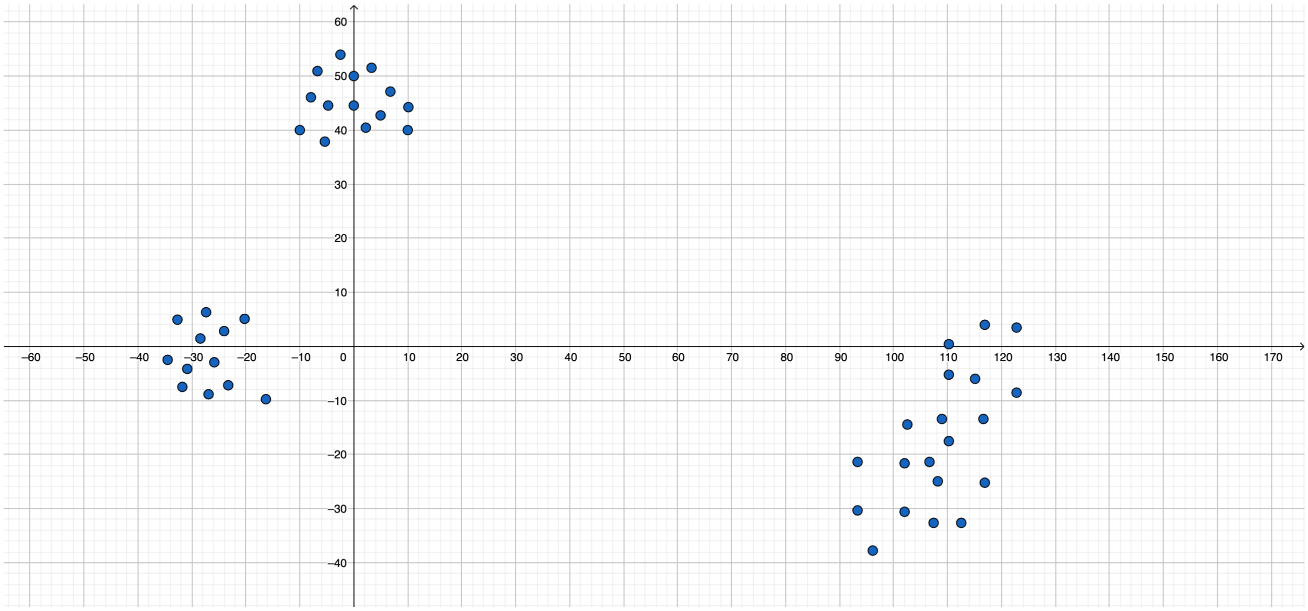
1. What are the cluster assignments?
2. What are the recomputed cluster centers?

3. Draw the cluster assignments after the first iteration on the graph below.



3.2 The importance of initialization

Given the points in the graph below, and assume we will have $k = 3$ cluster centers.



1. Give an example of a set of initialization points such that the K-Means algorithm would converge to a global minimum.

2. Give an example of a set of initialization points such that the K-Means algorithm would converge to a local minimum instead of the global minimum.