

Data Scientist Interview Preparation

Ce Peng

June 22 2021

Latex code ref <https://www.overleaf.com/learn>

Algorithm content ref <https://stanford.edu/~cpiech/cs221/handouts/kmeans.html>

1 Questions 1

What are drawback of K-means clustering?

- **Outliers** need to be removed before clustering as it can tend to include outliers into clusters.

1.1 Outlier

In [Wikipedia](#), it says that in statistics, an outlier is a data point that differs significantly from other observations.[1][2] An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the data set. An outlier can cause serious problems in statistical analyses.

the measurement or it may indicate experimental error; the latter are sometimes excluded from the data set. An outlier can cause serious problems in statistical analyses.

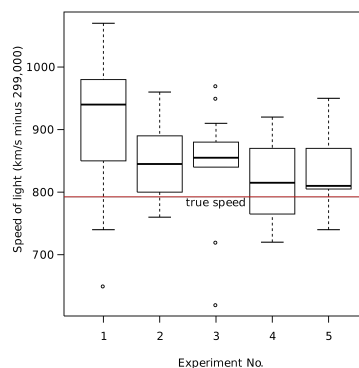


Figure 1: Box plot of data from the Michelson–Morley experiment displaying four outliers in the middle column, as well as one outlier in the first column.

1.2 K-means

This chapter is written by Chris Piech. Based on a handout by Andrew Ng.

1.2.1 Basic Idea

Say you are given a data set where each observed example has a set of features, but has no labels. Labels are an essential ingredient to a supervised algorithm like Support Vector Machines, which learns a hypothesis function to predict labels given features. So we can't run supervised learning. What can we do?

One of the most straightforward tasks we can perform on a data set without labels is to find groups of data in our dataset which are similar to one another – what we call clusters.

K-Means is one of the most popular "clustering" algorithms. K-means stores k centroids that it uses to define clusters. A point is considered to be in a particular cluster if it is closer to that cluster's centroid than any other centroid.

K-Means finds the best centroids by alternating between (1) assigning data points to clusters based on the current centroids (2) choosing centroids (points which are the center of a cluster) based on the current assignment of data points to clusters.

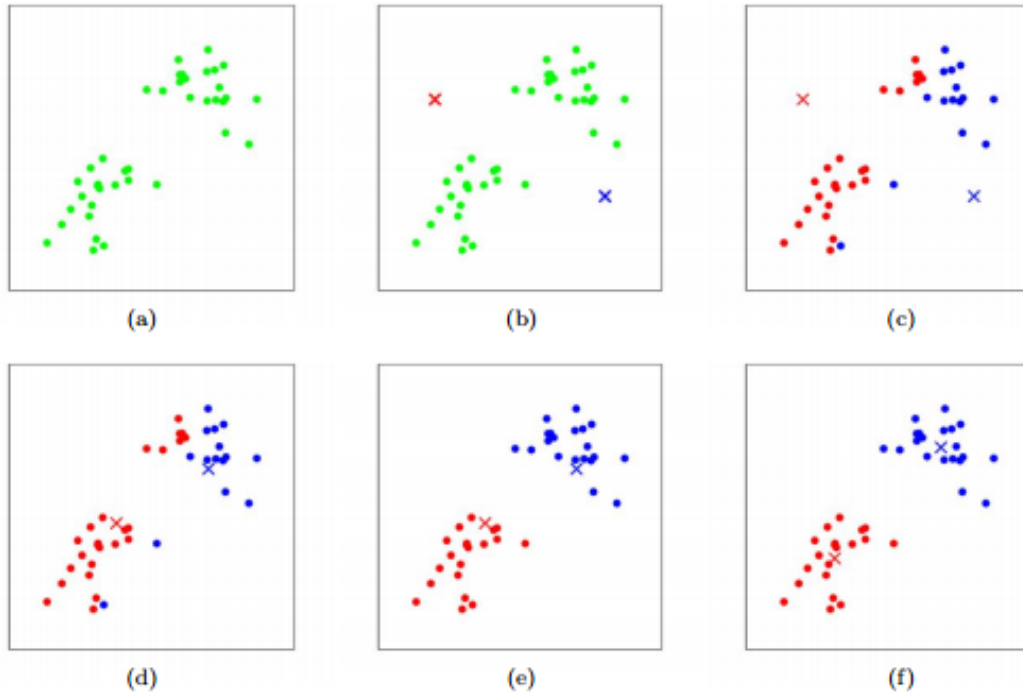


Figure 2: K-means algorithm. Training examples are shown as dots, and cluster centroids are shown as crosses. (a) Original dataset. (b) Random initial cluster centroids. (c-f) Illustration of running two iterations of k-means. In each iteration, we assign each training example to the closest cluster centroid (shown by “painting” the training examples the same color as the cluster centroid to which is assigned); then we move each cluster centroid to the mean of the points assigned to it. Images courtesy of Michael Jordan.

1.2.2 Algorithm

In the clustering problem, we are given a training set $x^{(1)}, \dots, x^{(m)}$, and want to group the data into a few cohesive “clusters.” Here, we are given feature vectors for each data point $x^{(i)} \in \mathbb{R}^n$ as usual; but no labels $y^{(i)}$ (making this an unsupervised learning problem). Our goal is to predict k centroids and a label $c^{(i)}$ for each datapoint. The k-means clustering algorithm is as follows:

1 Initialize **cluster centroids** $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^n$ randomly.

2 Repeat until convergence: {

For every i , set

$$c^{(i)} := \arg \min_j \|x^{(i)} - \mu_j\|^2.$$

For every j , set

$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

}

1.2.3 Implementation

Here is pseudo-python code which runs k-means on a dataset. It is a short algorithm made longer by verbose commenting.

```

1 # Function : K-means
2 # -----
3 # K-means is an algorithm that take in a dataset and a constant
4 # K and returns K centroid (Which define clusters of data in the
5 # dataset which are similar to one another)
6 def kmeans(dataSet, k):
7     # Initialize centroid randomly
8     numFeatures = dataSet.getNumFeatures()
9     centroids = getRandomCentroids(numFeatures, k)
10    # Initialize book keeping vars
11    iterations = 0
12    oldCentroids = None
13    # Run the main k-means algorithm
14    while not shouldStop(oldCentroids, centroids, iterations):
15        # Save old centroids for convergence test. Book keeping.
16        oldCentroids = centroids
17        iterations += 1
18
19        # Assign labels to each datapoint based on centroids
20        labels = getLabels(dataSet, centroids)
21
22        # Assign centroids based on datapoint labels
23        centroids = getCentroids(dataSet, labels, k)
24
25    # We can get the labels too by calling getLabels(dataSet, centroids)
26    return centroids

```

Listing 1: K-means Pseudo

```

1 # Function: Should Stop
2 # -----
3 # Returns True or False if k-means is done. K-means terminates either
4 # because it has run a maximum number of iterations OR the centroids
5 # stop changing.
6 def shouldStop(oldCentroids, centroids, iterations):
7     if iterations > MAX_ITERATIONS: return True
8     return oldCentroids == centroids

```

Listing 2: Should Stop Function Pseudo

```

1 # Function: Get Labels
2 # -----
3 # Returns a label for each piece of data in the dataset.
4 def getLabels(dataSet, centroids):
5     # For each element in the dataset, chose the closest centroid.
6     # Make that centroid the element's label.

```

Listing 3: Get Labels Function

```

1 # Function: Get Centroids
2 # -----
3 # Returns k random centroids, each of dimension n.
4 def getCentroids(dataSet, labels, k):
5     # Each centroid is the geometric mean of the points that
6     # have that centroid's label. Important: If a centroid is empty (no points have
7     # that centroid's label) you should randomly re-initialize it.

```

Listing 4: Get Centroid Function

Important note: You might be tempted to calculate the distance between two points manually, by looping over values. This will work, but it will lead to a slow k-means! And a slow k-means will mean that you have to wait longer to test and debug your solution.

Let's define three vectors:

```

1 x = np.array([1,2,3,4,5])
2 y = np.array([8,8,8,8,8])
3 z = np.ones((5,9))

```

To calculate the distance between x and y we can use:

```

1 np.sqrt(sum(x - y)**2)

```

To calculate the distance between all the length 5 vectors in z and x we can use:

```
1 np.sqrt(((z-x)**2).sum(axis=0))
```

1.2.4 Intuition

Figure 1 shows k-means with a 2-dimensional feature vector (each point has two dimensions, an x and a y). In your applications, will probably be working with data that has a lot of features. In fact each data-point may be hundreds of dimensions. We can visualize clusters in up to 3 dimensions (see figure 3) but beyond that you have to rely on a more mathematical understanding.

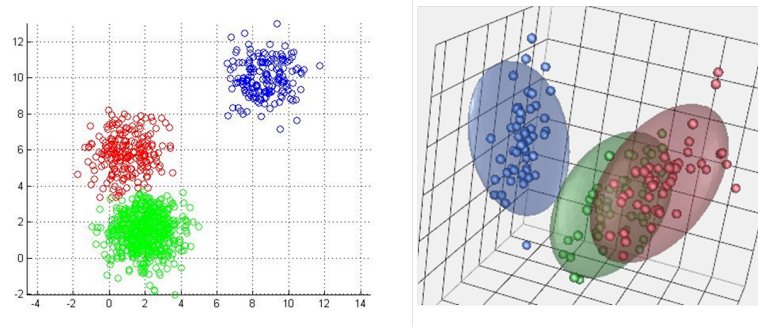


Figure 3: KMeans in other dimensions. (left) K-means in 2d. (right) K-means in 3d. You have to imagine k-means in 4d.

1.3 Expectation Maximization

K-Means is really just the EM (Expectation Maximization) algorithm applied to a particular naive bayes model.

To demonstrate this remarkable claim, consider the classic naive bayes model with a class variable which can take on discrete values (with domain size k) and a set of feature variables, each of which can take on a continuous value (see figure 2). The conditional probability distributions for $P(f_i = x|C = c)$ is going to be slightly different than usual. Instead of storing this conditional probability as a table, we are going to store it as a single normal (gaussian) distribution, with it's own mean and a standard deviation of 1. Specifically, this means that: $P(f_i = x|C = c) \sim \mathcal{N}(\mu_{c,i}, 1)$

Learning the values of $\mu_{c,i}$ given a dataset with assigned values to the features but not the class variables is the provably identical to running k-means on that dataset.

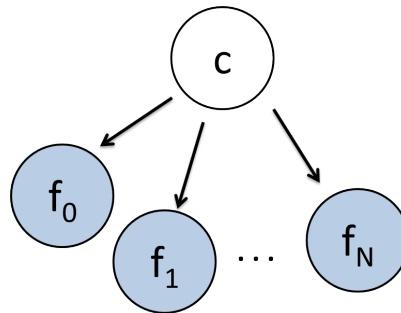


Figure 4: K-Means algorithm is the EM algorithm applied to this Bayes Net.

If we know that this is the structure of our bayes net, but we don't know any of the conditional probability distributions then we have to run Parameter Learning before we can run Inference.

In the dataset we are given, all the feature variables are observed (for each data point) but the class variable is hidden. Since we are running Parameter Learning on a bayes net where some variables are unobserved, we should use EM.

Lets review EM. In EM, you randomly initialize your model parameters, then you alternate between (E) assigning values to hidden variables, based on parameters and (M) computing parameters based on fully observed data.

E-Step: Coming up with values to hidden variables, based on parameters. If you work out the math of choosing the best values for the class variable based on the features of a given piece of data in your data set, it comes out to "for each data-point, chose the centroid that it is closest to, by euclidean distance, and assign that centroid's label." The proof of this is within your grasp! See lecture.

M-Step: Coming up with parameters, based on full assignments. If you work out the math of choosing the best parameter values based on the features of a given piece of data in your data set, it comes out to "take the mean of all the data-points that were labeled as c."

So what? Well this gives you an idea of the qualities of k-means. Like EM, it is provably going to find a local optimum. Like EM, it is not necessarily going to find a global optimum. It turns out those random initial values do matter.

2 Question 2

What is the criteria to split the node in decision tree?

- Gini Index

2.1 Decision Tree

2.2 Random Forest

3 Question 3

For medical equipment manufacture industry, does a blood pressure gauge equipment model having a low sensitivity and high specificity , a good model?

- **False** A High Sensitivity will ensure model does not have False Negative. A BP Patient will not be shown as not have BP as it can be more dangerous than having False Positive - a Non BP person shown as having high BP.

3.1 Sensitivity and Specificity

4 Question 4

In linear regression equation $\text{sales} = 30 \times \text{promotion} + 10 \times \text{distribution} + 4$. How much will sales increase for 1\$ increase in promotion keeping distribution constant?

- **30** As per Linear model interpretation, any co-efficient value or slope will determine the increase in target(sales) given 1 unit increase in feature keeping other feature constant. Example: 1\$ increase in promotion will cause sales to grow by approximately 30\$ keeping other feature constant.

5 Question 5

Which are anomaly detection methods or algorithms?

- **Tukey IQR** Tukey method of detecting univariate data anomalies where in any value below $Q1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$ gives a anomalies. DBSCAN also is a clustering method to detect the multivariate data anomalies. General method of anything above $Mean + Standard Deviation \times 3$ or $Mean - Standard Deviation \times 3$ are anomalies.

5.1 Anomaly Detection

5.2 K Nearest Neighbor

5.3 XGBoost

5.4 Density-based spatial clustering of applications with noise (DBSCAN)

6 Question 6

In Pandas `df.corr()` , where df is data frame is correct way to find correlation matrix?

- **True** `df.corr()` will give correlation matrix. Correlation can be between +1 to -1. A value greater than 0.7 or less than -0.7 are generally considered as high correlation between two variables.

7 Question 7

Two variables pressure and temperature have a correlation of - 0.8 what does it mean?

- **Pressure decrease with increase in temperature** They are inversely proportional . They are strong negative correlated meaning pressure will decrease with increase in temperature and vice- versa.

8 Question 8

A dependent variable values are ranks represented as 1,2,3,4,5,6 best model to predict the rank can be built by (select choices)

- **Random Forest Classifier** Since target is discrete or categorical hence classifier algorithms will be used.

8.1 Random Forest Regress-or

9 Question 9

Misclassification can be best possibly mitigated by using these options?

- **Stratified Sampling** Stratified sampling will split data as per strata and ratio of class for training and validation. Removing very less frequent class which are 1-5% are an potential option in case of multinomial classification only.

10 Question 10

MAPE is metric used to calculate?

- **Regression Error** MAPE refers to Mean Absolute Percentage error defined as : $Mean of \frac{abs((Actual - Predicted))}{Actual} \times 100$, the $\frac{abs((Actual - Predicted))}{Actual} \times 100$ is called percentage error and it shows how much percentage difference is present between actual value and predicted value

11 Question 12

Which of the options represent the outcome of increasing the eps value in DBSCAN?

- **Decrease Clusters** EPS acts as radius of circle of acceptance. In DBSCAN increasing EPS will cause to increase the boundary of selection or acceptance by increasing the circle circumference. Hence most of points will come under single or same clusters and hence anomalies or independent clusters (-1 numbered clusters) will be less or decrease.

12 Question 13

What is True about Random Forest?

- **Bagging Algorithm** Random forest is ensemble of Trees with bagging technique. Bagging is random sampling of data thereby reducing bias/variance or prevent underfit / overfit.

12.1 Bagging Algorithm

13 Question 14

What's the difference between Type I and Type II error?

- **Type I error is a false positive, while Type II error is a false negative.** Type I error is a false positive, while Type II error is a false negative. A clever way to think about this is to think of Type I error as telling a man he is pregnant, while Type II error means you tell a pregnant woman she isn't carrying a baby.

14 Question 15

Which component does Time series data decomposition provides:

- **Trend, Seasonality and Noise** Decomposition of time series gives trend components, seasonal components and noise(residue) components.

15 Question 16

Select the statistical test to understand whether time series is stationary?

- **Dickey-Fuller Test** The null hypothesis of Dickey Fuller test is that there is a unit root in an AR model, which implies that the data series is not stationary. Null Hypothesis (H0): If failed to be rejected, it suggests the time series has a unit root, meaning it is non-stationary. It has some time dependent structure. If p-value > 0.05 : Fail to reject the null hypothesis (H0), the data has a unit root and is non-stationary. If p-value ≤ 0.05 : Reject the null hypothesis (H0), the data does not have a unit root and is stationary.

16 Question 17

Which type of time series is the figure shown?

- **Multiplicative** In the multiplicative model, the original time series is expressed as the product of trend, seasonal and irregular components. You can observe the seasonal amplitude is increasing with time.

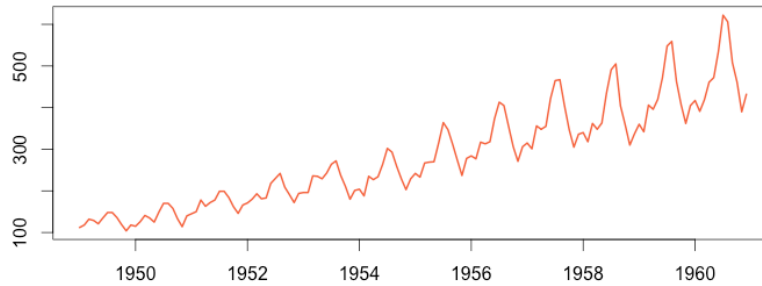


Figure 5: Data Curve

17 Question 18

Why do we need to Normalize the Data?

- The goal of **normalization** is to change the values of numeric columns in the dataset to use a common scale, without differences in the ranges of values or losing information. "Normalization as mentioned changes values to common scale by formula $\text{normalized value} = (\text{value} - \text{mean}) / \text{standard deviation}$ "

18 Question 19

Consider two records/rows with features as X_1, X_2 . record1 coordinates for $X_1 = 3$ and $X_2 = 4$ and record2 coordinates for $X_1 = 4$ and $X_2 = 6$, the Euclidian distance between them is:

- $\text{SQRT}(5) \dots$ means square root of 5. "Square Root of $(\text{Square}(3-4) + \text{Square}(4-6)) = \text{Square Root of } (\text{Square}(-1) + \text{Square}(-2)) = \text{Square Root of } (1 + 4) = \text{Square Root of } 5$ can be represented as below = $\text{SQRT}(5)$ "

19 Question 20

Normal Distribution can be observed by , Select the best options:

- Shapiro Wilk Test, Normality Test, Histogram, Box Plot "Shapiro Wilk Test or Normality Test Refer <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.shapiro.html> The Shapiro-Wilk test tests the null hypothesis that the data was drawn from a normal distribution."

20 Data Scientist Work Process



Figure 6:
ML Stages