Reading 8

☆ Article: Clustering

Clustering: automatically grouping together data points with similar characteristics and assigning them to "clusters"

- Clustering is unsupervised.

K-means Clustering

- better on larger datasets
- seperates data into several clusters, characterized by their centroids (midpoints)

- **Steps:**
  1. Select K, the number of clusters
  2. Randomly pick K existing points on your chart. These will be the centroids of the initial clusters.
  3. Measure the distance between each data point and each centroid and assign each data point to its closest centroid and the corresponding cluster.
  4. Recalculate the midpoint (centroid) of each cluster.
  5. Repeat steps 3 & 4 to reassign data points to clusters based on the new centroid locations. Stop when either:
     a) centroids have stabilized (no data points are reassigned)
     b) the predefined max number of iterations is reached.

- To find a good value of K, you can use an elbow plot.

  - elbow plots graph the variance for different values of K and then the inflection point on the curve is usually a good value for K.

    ↳ the variance is typically the sum of the squared distances from each point to its cluster center.

- Hierarchial Clustering

  - better on smaller datasets
  - clusters are assigned based on hierarchial relationships between data points
  - Two types of hierarchial clustering:

    1. agglomerative (bottom-up) more commonly used
    2. divisive (top-down)

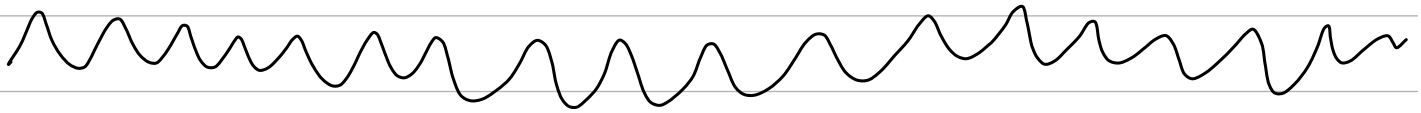
Agglomerative Steps:

1. Assign each data point to its own cluster, so the number of initial clusters (K) is equal to the number of initial data points (N).
2. Compute distances between all clusters.
3. Merge the two closest clusters.
4. Repeat steps iteratively until all data points are finally merged into one cluster.

- With divisive you would start with all data points in one cluster and split until we ended up with each datapoint as its own cluster.

- To measure similarity between two clusters we use linkage methods. They impact the results of hierarchial clustering. Some popular ones are:

1. Complete Linkage: uses maximum distance between any 2 points in each cluster.
2. Single Linkage: uses minimum distance between two points in each cluster.
3. Average Linkage: uses average distance between each point in each cluster.



☆ Video: Principal Component Analysis (PCA) using Singular Value Decomposition (SVD)

- Dataset: We've measured the transcription of two genes (gene 1 and gene 2) in 6 different mice.

- PCA can take 4 or more dimensions of data, and can make a 2D PCA plot.
  ↳ This plot will show us that similar mice cluster together.

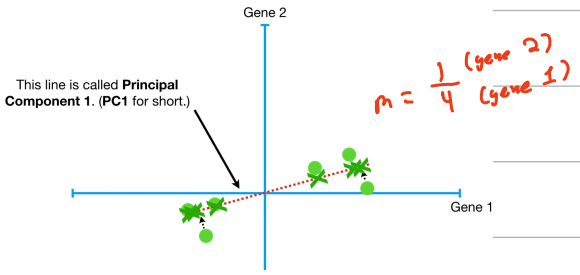
- PCA can also tell us which variable is most valuable for clustering the data.
- PCA can also tell us how accurate the 2D graph is.

- In our 2 gene example we:

1. Caculate average gene 1 value
2. Calculate average gene 2 value
3. plot the average point
4. Shift the average point to the origin (all data points move relative to center)
5. Now we fit a line through the data (line must go through origin) called PC1.

↳ PCA finds the best line by maximizing the sum of the squared distances from the projected points to the origin.

↳ This line is called Principal Component 1 (PC1)

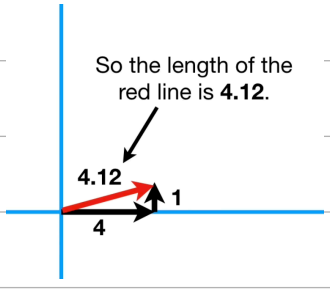


↳ The slope can tell us what feature has more spread.

↳ In our example the slope tells us that Gene 1 is more important when it comes describing how our data is spread out

↳ PC1 is a linear combination of gene 1 and gene 2

↳ You use pythagorean thm to find the hypotnuse of the slope



↳ When you do PCA with SVD, the ratio (slope) is scaled so that the hypotnuse equals one. (So divide each side by length of hypotnuse)

↳ So we get 0.97 for Gene 1 and 0.242 for Gene 2

↳ The vector <gene 1, gene 2> or < 0.97, 0.242 > is called the singular vector or eigenvector for PC1.

↳ And the proportion of each gene are called loading scores.

↳ Also the sum of squared distances (to the origin) for the best fit line is the eigenvalue for PC1.

↳ The square root of that value is called the singular value for PC1.

6. Then we find PC2

↳ Since our data is only 2D, PC2 is simply the line orthogonal to PC1.

↳ So its vector consists of -1 for Gene 1 and 4 for gene 2 (original scaling)

↳ < -0.242, 0.97 > (scaled)

↳ This is the singular vector or eigenvector for PC2.

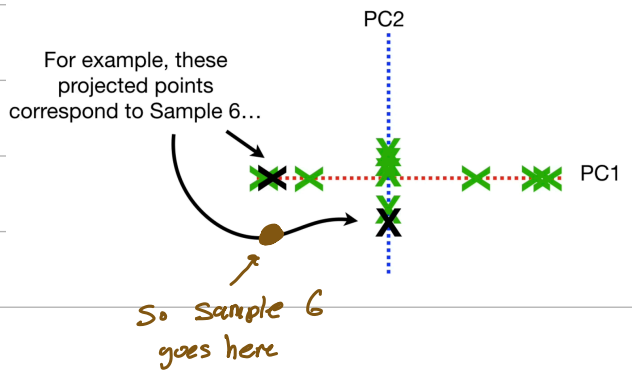
↳ -0.242 and 0.97 are the loading scores for PC2.

↳ They tell us Gene 2 is 4 times more important than Gene 1 for PC2.

↳ The eigenvalue for PC2 is the average of the sum of squares of the distances between the projected points and the origin.

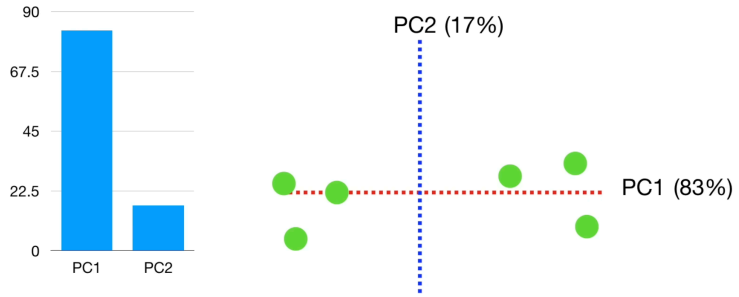
7. To draw the final PCA plot we simply rotate everything so that PC1 is horizontal.

↳ Then we use the projected points to figure out where our points go:



- The eigenvalues of PC1 and PC2 are just measures of variance in this context.

- A scree plot is a graph representation of the percentages of variation each PC accounts for:



- PCA with 3 variables is pretty much the same as with 2 variables.
  - ↳ In this we will have another line called PC3 that goes through the origin and is orthogonal to both PC1 and PC2.

- In practice the number of PCs is either the number of variables or the number of samples, whichever is smaller.

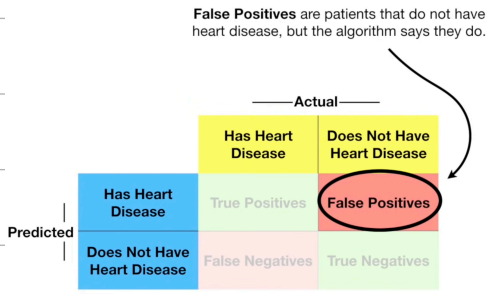
- To generate the PCA plot, you simply plot the two dimensions that account for the most variance in the scree plot. (Strip away all other features)
  - ↳ If the two bars sum to 94%, then 94% of the total variance is captured in your 2D PCA plot.

- Note! If the bars in the scree plot are all similar height, then the PCA plot will not be as useful.

☆ Video: Confusion Matrix

- A way to evaluate machine learning models (classification tasks)

- Ex:



- The green boxes (along the main diagonal) correspond to correctly classified samples.
- The red boxes correspond to incorrectly classified samples.
  - ↳ This scales up to higher dimension confusion matrices too.
  - ↳ The size of the confusion matrix is determined by the number of things you want to predict.
    - ↳ If you have x classes to choose from for your prediction then you get an x by x confusion matrix.

☆ Video: Precision vs Recall

- Precision and Recall are classification metrics.

- Accuracy has some issues:
  - ↳ especially with imbalanced classes.

- A model predicts apple or orange (with a line):
  - ↳ To calculate precision for apple class:
    - ↳ Look at only apple side of model (ignore other side)
    - ↳ $\frac{\text{total \# of apples correct}}{\text{total observations on apple-side}}$
  - ↳ To calculate recall for apple class:
    - ↳ focus in on all the actual apples:
    - ↳ $\frac{\text{\# total apples correct}}{\text{\# total actual apples}}$

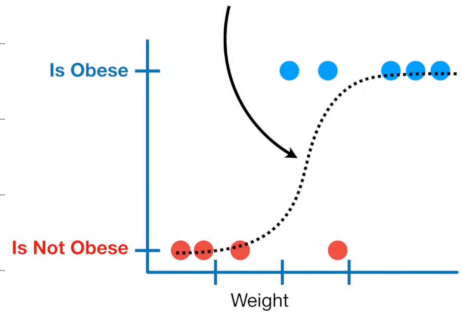
- These can guide the model to improve its decision boundry.

- An F_1 score is the harmonic average of precision and recall.

☆ Video: ROC and AUC

- Our data has two classes obese and not obese.
  - ↳ Predictions are based on weight.

- We fit a logistic regression curve to the data:



- The curve tells us the probability that a mouse is obese based on its weight.
  - ↳ Then we set a threshold for classification (0.5 for our example)

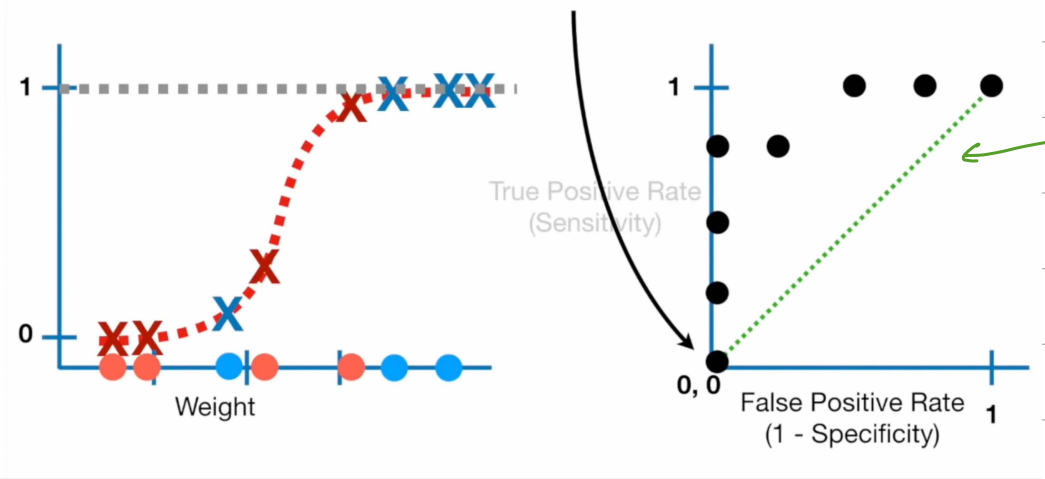
- You can create a confusion matrix based on classifications.
  - ↳ Then you can calculate sensitivity and specificity.

- The threshold could be set to anything between 0 and 1.
  - ↳ How do we determine which threshold is the best?

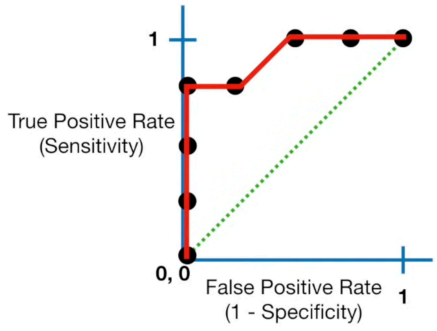
- We can use Reciever Operator Characteristic (ROC) graphs
  - ↳ The y-axis shows the true positive rate (sensitivity): $\frac{\text{true positives}}{\text{true positives + false negatives}}$
  - ↳ The x-axis shows the false positive rate (1 - specificity): $\frac{\text{false positives}}{\text{false positives + true negatives}}$

- Ex: ROC Plot

each point corresponds to a different decision boundry

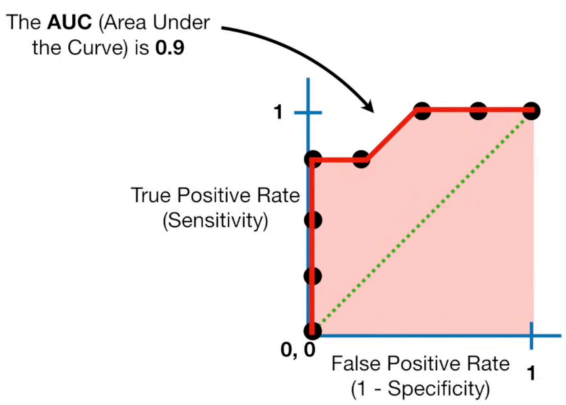


The green line shows where the true positive rate equals false positive rate



The ROC graph summarizes all of the confusion matrices the each threshold (decision bondry) produced.

- AUC (Area under curve)



- The AUC makes it easy to compare one ROC curve to another.

- The greater the AUC is better the model.
  ↳ (you have one ROC plot per model)

- Note: Sometimes the false positive rate is replaced with precision.