Toggle code

drawing

Week 10 - Clustering

Dr. David Elliott

1. Preprocessing
2. Data Exploration and Segmentation
3. Semi-Supervised Learning
4. Strengths and Limitations

Notes

- Clustering has a vast range of use cases, of which are few are going to touch on a few explored in this lecture/notebook.

6. Preprocessing

Dimensionality Reduction^2

One clustered, we can measure the *affinity* of each instance with its cluster (how well it fits in the cluster).

Cluster affinities can be used as features (e.g. a K dimensional vector), often leading to a reduced feature space than the feature vector - but still preserving some information.

Example: Image Classification^2

The simple MNIST-like dataset availble in Sklearn is a common classification example.

Notes

- The example contains 1,797 grayscale 8×8 images representing digits 0 to 9.
- The example below clusters the training set into 50 clusters and replace the images with their distances to the 50 clusters
- As k here is a preprocessing step in a classification pipeline you can use search for the "optimal" value of k

```
Logistic Regression Accuracy: 96.89%
KMeans and Logistic Regression Accuracy: 97.78%
```

7. Data Exploration and Segmentation

Data Exploration

Clustering is useful for when we are looking for new patterns in data.

It maybe we want to cluster our data into groups, and analyze each cluster separately.

Example: Cancer Research[1]

A researcher might assay gene expression levels in patients with breast cancer.

They might look for subgroups among the breast cancer samples, or among the genes, in order to obtain a better understanding of the disease.

Notes

- *"For instance, suppose that we have a set of n observations, each with p features. The n observations could correspond to tissue samples for patients with breast cancer, and the p features could correspond to measurements collected for each tissue sample; these could be clinical measurements, such as tumor stage or grade, or they could be gene expression measurements. We may have a reason to believe that there is some heterogeneity among the n tissue samples; for instance, perhaps there are a few different unknown subtypes of breast cancer. Clustering could be used to find these subgroups. This is an unsupervised problem because we are trying to discover structure—in this case, distinct clusters—on the basis of a data set."*[1]

Recommender Systems

In a very general sense, recommender systems are algorithms aimed at suggesting relevant items to users (e.g. movies, text, products, ect.)[3].

Example: Customer Segmentation[2]

You can cluster your customers based on their purchases and website activity.

It is useful to know who your customers are you can adapt products and marketing campaigns.

Notes

- An online shopping site might try to identify groups of shoppers with similar browsing and purchase histories, and items of particular interest to the these shoppers. Then an individual shopper can be preferentially shown the items in which he or she is particularly likely to be interested, based on the purchase histories of similar shoppers[1].
- For hierarchical models the choice of dissimilarity measure here is very important, as it has a strong effect on the resulting dendrogram. For instance, the type of dissimilarity measure that should be used to cluster the shoppers depends on the goal. If Euclidean distance is used, then shoppers who have bought very few items overall (i.e. infrequent users of the online shopping site) will be clustered together. If correlation-based distance is used, then shoppers with similar preferences (e.g. shoppers who have bought items A and B but never items C or D) will be clustered together, even if some shoppers with these preferences are higher-volume shoppers than others[1].
  - This will also be effected by whether the variables are scaled to have standard deviation one before the dissimilarity between the observations is computed. For example, some items may be purchased more frequently than others; for instance, a shopper might buy ten pairs of socks a year, but a computer very rarely. High-frequency purchases like socks therefore tend to have a much larger effect on the inter-shopper dissimilarities, and hence on the clustering ultimately obtained, than rare purchases like computers. This may not be desirable[1].
  - We might also want to scale the variables to have standard deviation one if they are measured on different scales; otherwise, the choice of units (e.g. centimeters versus kilometers) for a particular variable will greatly affect the dissimilarity measure obtained.
  - The issue of whether or not to scale the variables before performing clustering applies to K-means clustering as well[1].
- If your interested in trying this out, this data set may be useful.
  - Two example projects using this data:
  - https://www.kaggle.com/hellbuoy/online-retail-k-means-hierarchical-clustering
  - https://www.kaggle.com/tklimonova/online-retail-cohort-and-rfm-analysis

Extra Example: Search Recommendations[1,2]

A search engine might choose what search results to display to a particular individual based on the click histories of other individuals with similar search patterns.

Some search engines allow you to search for images similar to a reference image.

Notes

- You may first build a clustering algorithm to images in a database, then similar images end in similar clusters allowing you to return images from the same cluster.[2]

Image Segmentation

An image can be segmented in a number of different ways:

- Semantic Segmentation
  - All pixels part of the same *object type* get assigned to the same segment
- Instance Segmentation
  - All pixels part of the same *object* get assigned to the same segment

Example: Color Segmentation

By clustering pixels acording to color, then replacing each pixel with the mean cluster color, you can reduce the number of colors in an image[2].

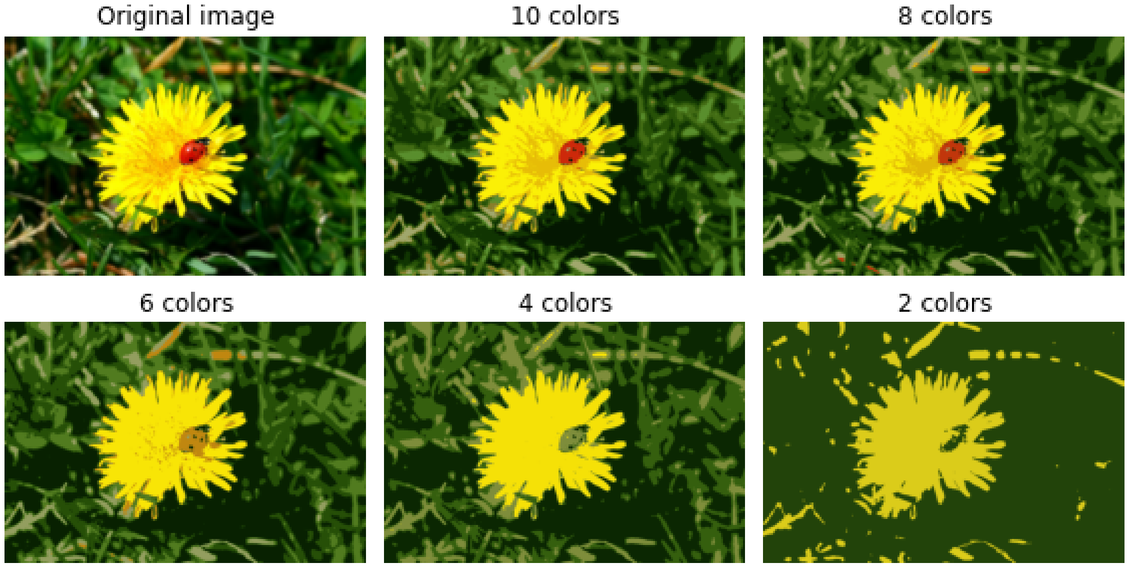
This is used in many object detection and tracking systems as contours of objects become more defined[2].

Notes

- An example of Semantic Segmentation would be that for a self-driving cars vision system, all pixels belonging to pedestrians are clustered[2].
- An example of Instance Segmentation is that there would be a separate cluster for each individual pedestrian[2].
- Semantic and Instance Segmentation is typically done using convolutional neural networks.

- Another example of Color Segmentation is the use of satellite images to measure total forest area.

```
(533, 800, 3)
```



Notes

- In the color segmentation example above, we are using the k-means centers to replace the clusters color.
- K-means prefers similar sized clusters, which is why the ladybugs red color dissapears below 8 colors.

8. Semi-Supervised Learning

Semi-supervised learning is used in the case where we have lots of unlabelled examples, and few labeled cases (common).

Example: Building an MNIST-like Dataset[2]

Returning to the MNIST-like dataset, we got 96.89% accuracy on a test set (25% of the full data) when we trained a logistic regression using the rest (75%).

Imagine we only had our test set that we have gone through and labeled, and now need to label our training set.

Instead of labelling all 1347 instances manually, we could start by training a k-means algorithm and just manually labeling instances closest to the centeroid for each cluster.

```
KMeans 50 Labels and Logistic Regression Accuracy: 92.22%
```

Notes

- A classifier where we only had to label 50 observations instead of the 1347 in the previous training set getting ~5% less accuracy is pretty good.

We could further improve performance by using these labels we made to automatically label a percentage of the observations close to the centeroid.

```
Propogated Label Accuracy: 97.51%
Propogated KMeans and Logistic Regression Accuracy: 93.56%
```

Active Learning[2]

To continue improving this model above is we could use active learning, where human experts provide labels for specific instances when requested by the algorithm.

A strategy for this is *uncertainty sampling*:

1. A model trains using labeled instances and makes predictions on unlabelled instance
2. The instances where the model is the most uncertain (e.g. using probabilities), is given an expert label.
3. You iterate the above process until performance improvement stops being worth the labelling effort.

Other strategies include:

- Labeling instances that will result in the most model change,
- Result in the largest drop in validation error,
- Instances where different models (e.g. SVM or Random Forest) disagree the most.

9. Strengths and Limitations

K-Means

Advantages

- k-means generally has linear computational complexity regarding the number of instances, n, number of clusters, k, and the number of dimensions d.
    - This is only true when there is a clustering structure.
    - K-Means is generally one of the fastest clustering algorithms.
- Clustering methods generally are not very robust to perturbations to the data[1].

Notes

- Perturbation Robustness: *"For instance, suppose that we cluster n observations, and then cluster the observations again after removing a subset of the n observations at random. One would hope that the two sets of clusters obtained would be quite similar, but often this is not the case!"*[1]

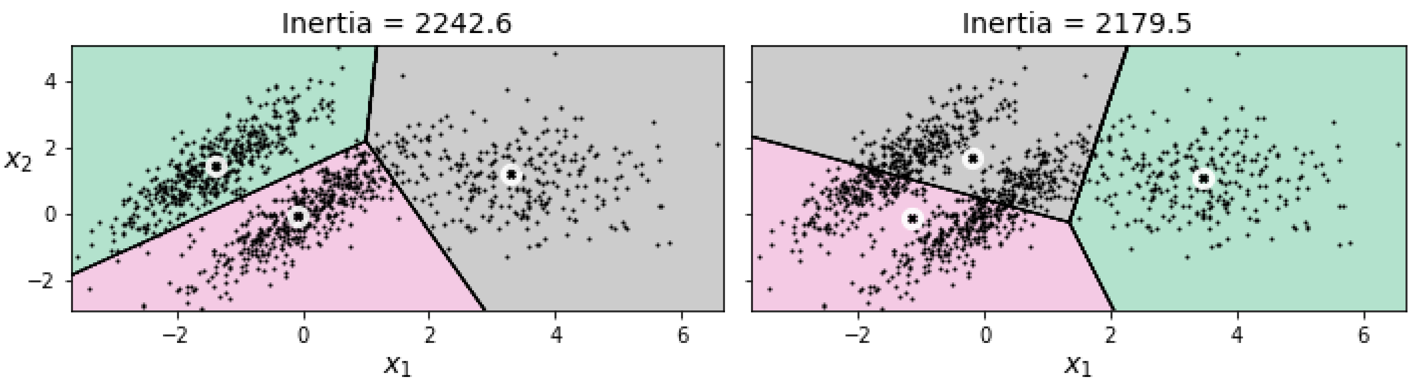
Disadvantages

- Does not work very well with clusters with different densities or non-spherical shapes[2].
- Requires us to pre-specify the number of clusters[1].
- Generally require scaling to be effective[1].
- Can result in empty clusters[4].
- Can have problems with convergence[4].
- Assumes that all the data belongs in clusters with no noise[5].

Notes

- Empty clusters are dealt with in Scikit-Learn by searching for the sample that is the fathest away from the centroid of an empty cluster, and reassign the centeroid to that.
- convergence problems can be dealt with by changing the tolerance for the changes in the within-cluster sum-squared-error to be declared as convergene (e.g. `tol=1e-04`)

Example of poor performance on ellipsoidal blobs



Hierarchical Clustering

Advantages

- Results in an interpetable tree-based representation of the observations (dendrogram)[1].
- Doesn't need the number of clusters to be pre-specified.
- Agglomerative clustering is generally stable across runs and the dendrogram shows how it varies over parameter choices[5].

Disadvantages

- Relies on careful selection of dissimilarity measure[1].
- Generally requires scaling to be effective[1].

- Assumes a hierarchical structure to the clusters, which may be unrealistic[1].
- Assumes that all the data belongs in clusters with no noise[5].

Notes

- *"The term hierarchical refers to the fact that clusters obtained by cutting the dendrogram at a given height are necessarily nested within the clusters obtained by cutting the dendrogram at any greater height. However, on an arbitrary data set, this assumption of hierarchical structure might be unrealistic. For instance, suppose that our observations correspond to a group of people with a 50–50 split of males and females, evenly split among Americans, Japanese, and French. We can imagine a scenario in which the best division into two groups might split these people by gender, and the best division into three groups might split them by nationality. In this case, the true clusters are not nested, in the sense that the best division into three groups does not result from taking the best division into two groups and splitting up one of those groups. Consequently, this situation could not be well-represented by hierarchical clustering. Due to situations such as this one, hierarchical clustering can sometimes yield worse (i.e. less accurate) results than K-means clustering for a given number of clusters."*[1]

DBSCAN

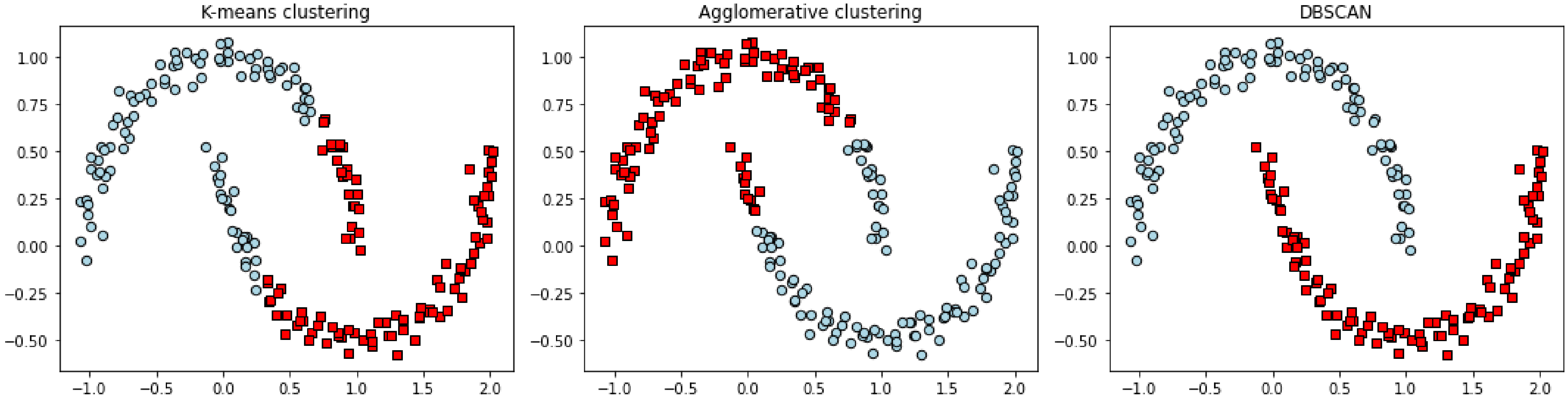
Advantages

- Can identify clusters of any shape[2].
- Is robust to outliers[2].
- Has only two hyperparameters (`eps` and `min_samples`)[2].
- Can be relatively linear to compute with number of instances; $O(m \log m)$[2].
  - Although can be $O(m^2)$ if `eps` is large.
- DBSCAN is generally stable across multiple runs[5]
  - But stability over varying `eps` and `min_samples` is not so good.

Disadvantages

- The algorithm can be quite sensitive to `eps` [5].
  - It can also be hard to pick in practice.
- Performs worse with in increasing number of features

- Although the same can be said for k-means and hierarchical methods too.



Extra

Other Clustering Algorithms[2]

`Scikit-Learn` implements many more clustering algorithms so here is a brief overview:

Affinity Propogation

Instances vote for similar instances to represent them, with representatives and votesrs forming a cluster upon conversion.

- Can find clusters of different sizes
- Not suited for large datasets

Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH)

Builds a minimal tree structure to quickly assign instances to clusters without having to store them all in memory.

- Designed for large datasets
- Can be faster, with similar results, to batch k-means provided features < 20.

Mean-Shift

Places a circle on each instance (the radius of the circle being a *bandwidth* hyperparamter), calculates the mean of all instances located within the circle, then shifts the circle so it is centred on the mean. This is iterated until all circles stop moving.

- Can find clusters of any shape
- Has few hyperparameters
- Chops clusters into pieces that have internal density variations
- Not suited for large datasets

Spectral Clustering

Creates a low-dimensional embedding for a similarity matrix between instances, and then uses another clustering algorithm (e.g. K-means).

- Can capture complex cluster structures
- Can be used to cut graphs
- Not suited for large datasets
- Poor performance when clusters have different sizes

References

1. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
2. Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media.
3. https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada
4. Raschka, S., & Mirjalili, V. (2019). Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing Ltd.
5. https://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html

```
[NbConvertApp] Converting notebook 3_Applications.ipynb to html
[NbConvertApp] Writing 799822 bytes to 3_Applications.html
[NbConvertApp] Converting notebook 3_Applications.ipynb to slides
[NbConvertApp] Writing 979089 bytes to 3_Applications.slides.html
[NbConvertApp] Converting notebook 3_Applications.ipynb to html
[NbConvertApp] Writing 1034526 bytes to PDF_Prep\3_Applications_no_code.html
```