ABSOLUTDATA

BRAINWAVE

DATA SCIENCE DIGEST

1ST EDITION

Q3 2019

Ina Nanda
Director, Market Research

Greetings,

A hearty welcome to the first public edition of BrainWave!

BrainWave is our new quarterly data science digest of articles from our data science, analytics and data engineering teams.

The vibrant and colorful summer season lends itself quite well to the theme for this first edition: Clustering and Segmentation. If we were to pick up a brush and paint people based on behavior, attitudes, trends and demographics, we would certainly have a kaleidoscope of hues and textures. When we look at this from a marketer's lens, we must understand where these colors merge versus where they stand out. Therein lies the art and science of segmentation.

A masterful segmentation result cannot be achieved overnight. One wishes it was as simple as opening a bottle and letting the genie out: poof, here are your segments. In reality, it's more like Pandora's box – the more you dig into the data, the higher your chances of getting lost in the data maze.

In today's data deluge, it's a delight to have tools and techniques to make sense of the many challenges that information overload throws our way. In this edition, our team has put together a suite of technical articles to help increase your knowledge and confidence in segmentation and clustering.

But what about the art of segmentation? From our experience, breathing life into your segments is just as important as the data munging. Capturing what's unique and telling about segments is the difference between internal adoption, inspiring new thinking, gaining insight, and capturing the personality of a segment. While this could be an entire book in itself, I'll focus here on two overarching principles we've seen that contribute heavily to segmentation success:

- Find actionable segments
- Ascribe insightful segment names

Actionable Segments

The basis for any segmentation should directly address and work towards supporting clear business objectives. Actionable segments capture a cross section of consumer data/behaviors/trends, industry dynamics, and your business objectives. For example:

- If the objective is an NPD, focus on differences in category needs (e.g. where customers purchase, what they look for in a product, response to pricing, etc.)
- If the objective is communication, focus on differences in brand offerings (e.g. customers' affinity towards specific brands, brand image/perceptions, source of brand awareness, etc.)
- If the objective is a brand extension, focus on product preferences (e.g. brands that customers currently prefer, flavor preferences, etc.)

To find actionable segments, it's important to find the skews and anomalies that offer deeper insights relevant to your business objectives. You can then showcase these segments to executives and other stakeholders, highlighting the unique characteristics of a segment. At the same time, it is equally important to consider the potential impact of your marketing activities on different segments – it is market reality that some segments will behave in a similar fashion to certain marketing inputs. Only when we've looked at the consumer lens and marketing lens, THIS is what makes the understanding of segments comprehensive.

So, how do you recognize these relevant skews, anomalies and response to marketing interventions? The answer is in the data. And here are some of the types of questions you should be exploring to find the color and flavor that will make a segment actionable. If you address at least one aspect from each of 3-5 key filters that align with your objectives, you should see themes emerge. For example:

- What are the category drivers for the segment? How do they respond to adjacent categories?
- What interests the segment about brands in the category?
- How does the segment interact with brands in various channels (shopping and media)?
- What are the psychographic differences and preferences?
- Assign other filters important to your business objectives

Apply each of your chosen filters for each initial segment that is uncovered. This exercise is often overlooked, and the result is predictable one-dimensional segments that are less likely to inspire action. However, understanding the similarities in addition to the differences is what actually brings the rigor and insight to your segments.

Insightful Segment Names

Then comes the most challenging part: thinking of both the overarching theme that reflects the story of your segments, AND naming each segment to bring life and color. This is where art meets science. To do this well, we must be aware of the latest jargon and trends, especially in the category of interest. The idea is to use a few well-chosen words that embody the attitudinal/behavioral and demographic skews of a particular segment.

A couple pearls of wisdom I can offer for this stage are to ensure that the chosen nomenclature is reflective of the core segment characteristics and that it succeeds in calling up a consistent image in most peoples' minds. Once this christening is done, the name becomes a buzzword in your corridors: marketing efforts are synced around these specific segments. And so begins a fresh new focus and understanding of your customer base.

Welcome to the "Clustering and Segmentation" edition of BrainWave. Happy reading and best of luck in your segmentation endeavors!

Ina Nanda
Practice Lead, Marketing Research at Absolutdata

Contents

Ensemble Segmentation

Statistictionary

Overview

Ensemble Segmentation, as the name suggests, implies combining several segmentation solutions that have been developed on the same data to create ensemble segments that best solve the problem at hand. Ensemble Segmentation first appeared in data mining literature in the mid-1990s.

Ensemble Segmentation is motivated by the fact that an ensemble is likely to be richer than its one-dimensional constituent solutions. In the context of market research, Ensemble Segmentation may mean combining segmentation solutions that have been developed independently from behavioral and attitudinal data on the same sample, thereby increasing our understanding of customers. In the context of CRM data, this may mean combining solutions that have been developed independently using marketing data and sales data, thereby allowing us to add customer intelligence to sales activity.

Typical Phases of Ensemble Segmentation

Phase 1: Develop multiple solutions that vary in terms of the method employed (K-Means, Hierarchical Clustering using the number of clusters (ranging from 2 to 30)) and the measures used. It is common to generate between 70 and 300 analyses during this phase.

Phase 2: Cluster and group respondents based on the analyses generated in Phase 1. Different meta-clustering algorithms can be used to cluster respondents; one approach is to use distance-based methods like K-Means.

Advantages of Ensemble Segmentation

There are some inherent advantages of creating an ensemble of segments:

1

Combining groupings from alternate and dissimilar sets of variables (e.g. demographics, lifestyle behaviors, desired benefits or needs, etc.) is likely to lead to richer insights.

2

Analysts can include a variety of clustering techniques when building the ensemble in Phase 1; it is not restricted to one approach.

3

Legacy clusters based on internal data can alsobe incorporated.

4

Cluster solutions that are less sensitive to sample variations and outliers are uncovered.

5

Solutions that would not have been obvious with a single approach become apparent.

Approaches to Ensemble Segmentation

There are several approaches that can be used in Ensemble Segmentation: **Bayesian methods using Gibbs Sampling, EM Algorithms, Hypergraph Partitioning, K-Means Clustering, Natural Clusters Combined from Shared Nearest Neighbors,** etc.

Reference for Further Reading

An Ensemble Method for Clustering -
Andreas Weingessel, Evgenia Dimitriadou & Kurt Hornik

-Authored by
Sunny Verma,
Data Scientist at Absolutdata

Prediction of Cluster Membership

Coder's Cauldron

Conventional Approach – Building a Typing Tool

The standard practice after carrying out a clustering exercise is to develop a 'typing tool' or classification model on the clustering solution to determine which cluster a new data point will belong to. There are numerous drawbacks of this orthodox practice:

1. The classification model is chosen based on the decision boundary, sample size, level of measurement of independent variables, etc. The underlying assumptions which a particular classification model follows also need to be met. Moreover, building a typing tool entails multiple iterations and an in-depth analysis of the degree of differentiation of variables across clusters, which is usually done manually and thus takes a lot of time.

2. Typing tools can suffer from lower predictability, inability to tolerate missing values and lack of robustness when applied to other samples.

Automated Approach – Scoring of Clustering Model

Since the clustering exercise can be materialized using various metrics like those of **similarity** (e.g. correlation), **compactness** (e.g. k-Means, mixture models), **connectivity** (e.g. spectral clustering) etc., it is very difficult for the analyst to figure out the optimal decision boundary for the classification model to use in order to predict the cluster membership.

The best recourse could be a readymade function capable of scoring the new examples into the identified clusters, based on the cluster boundaries formed through a particular clustering algorithm. This alternative will overcome the aforementioned limitations of typing tools as well. There are some inbuilt packages available in Python and R which can score the clustering model built with the algorithms of k-Means, kNN, SVM etc. You can find examples of this approach in the technical article on Spectral Clustering.

Implementation in Python/R

Prediction of k-Means Cluster Membership

One can make predictions based on new incoming data by calling the predict function of the k-Means instance and passing in an array of observations. The predict function calculates the distance of any new observation from all the k centroids/cluster centers of the k clusters. When the function finds the cluster center that the observation is closest to, it outputs the index of that cluster center's array. An implementation of predict in R can be found in the 'flexclust' package.

Prediction of kNN (k-Nearest Neighbors) Cluster Membership

kNN makes predictions using the training dataset directly. Predictions are made for a new instance (x) by searching through the entire training set for the k most similar instances (neighbors) and summarizing the output variable (e.g. Mode) for those k instances. To determine which of the k instances in the training dataset are most similar to a new input a distance measure (e.g. Euclidean distance for real-valued input variables) is used.

-Authored by

Pranav Sharma,
Data Scientist at Absolutdata

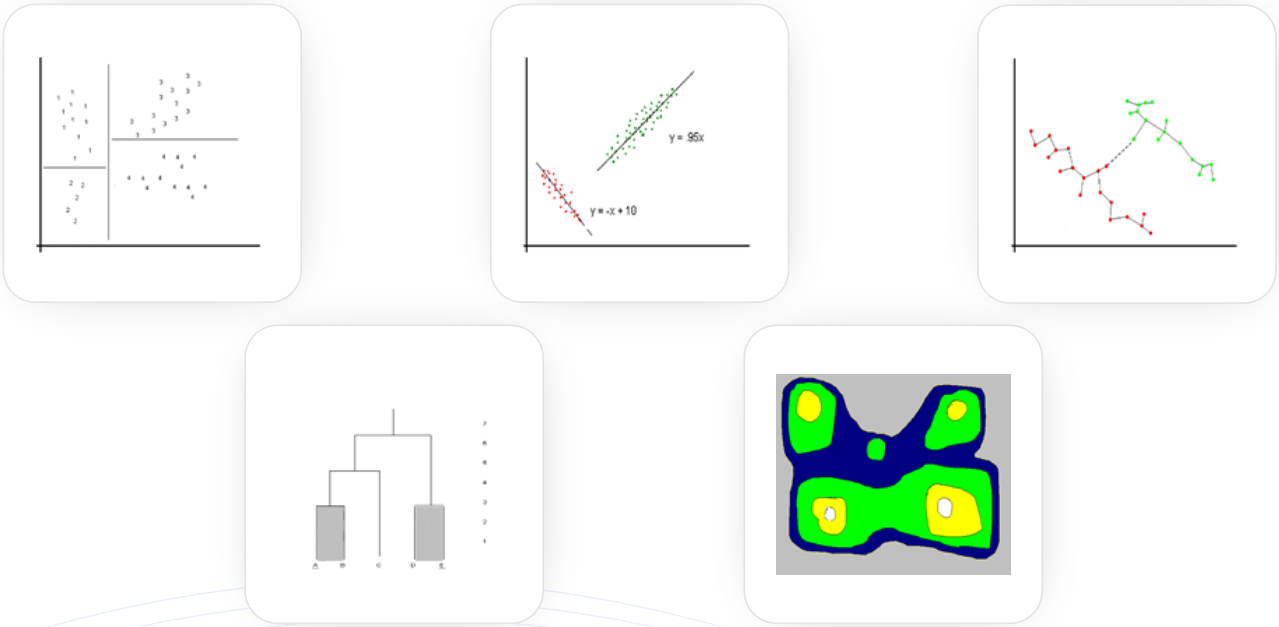
Pre-Processing & Post-Processing Clustering Visualization

Vivid Visualization

Overview

For analysts, the most common problem in clustering is arriving at the **inherent partitions** of a data set. In most algorithms' experimental evaluations, 2D-data sets are used so that the reader is able to visually verify the validity of the results (i.e. how well the clustering algorithm discovered the clusters of the data set).

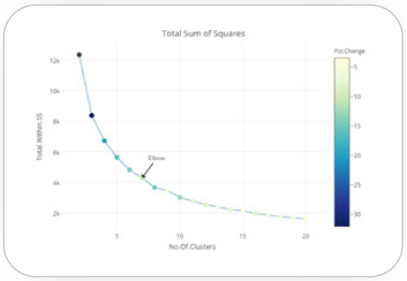
Clearly, data set visualization is a crucial verification of clustering results. In the case of large multidimensional data sets (i.e. more than three dimensions), effective visualization of the data set is difficult. Moreover, perceiving clusters using available visualization tools is hard for people unaccustomed to higher dimensional spaces. Some of the available data visualization options are **Scatter Plot, Minimum Spanning Tree, Dendrogram,** and Smoothed **Data Histogram.**

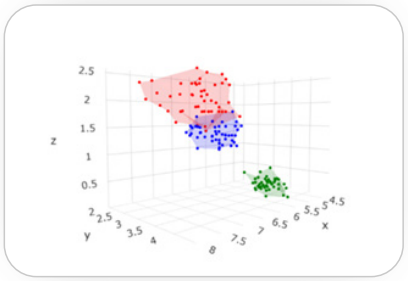


Data Visualization in Python

There are numerous libraries for data visualization in Python, including **matplotlib, Seaborn, ggplot, Bokeh , plotly, pygal, Altair, geoplotlib, Gleam, leather,** and **missingno.**

Plotly is a web-based data visualization platform for data scientists and engineers. The engine behind it is plotly.js, an open-source charting library built on D3.js and stack.gl. The following graphs represent the elbow curve and 3-D visualization of the clusters (K-means clustering) for the Iris dataset using plotly. Also, find below jupyter notebook with the implementation of the same in Python.





Over 2.5 quintillion bytes of data are created every single day. Not all the hidden patterns and trends can be identified just by going through millions of rows of data. Data visualization can affect an organization in both positive and negative ways. Improper visualization may bias your results and lead to faulty decisions whereas correctly applied data visualization can give clearer insights along with improving efficiency.

-Authored by
Manshi Poonia,
Data Scientist at Absolutdata

Spectral Clustering

Thriving Traction

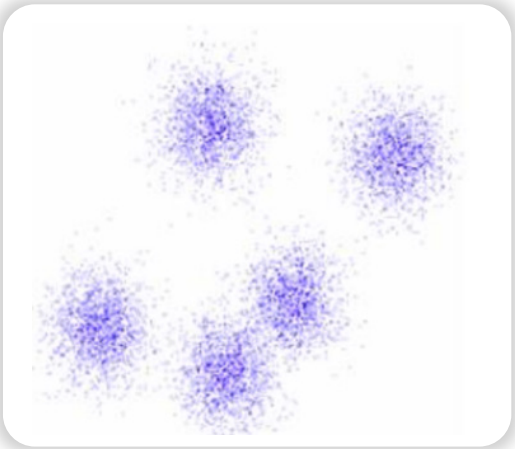
Overview

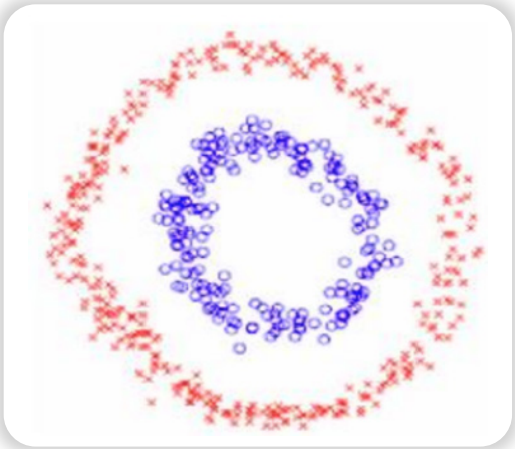
Clustering is a widely used unsupervised learning technique. The grouping is such that the points in a cluster are similar to each other. **'Spectral Clustering' uses the connectivity approach to clustering,** wherein communities of nodes (i.e. data points) that are connected or immediately next to each other are identified in a graph. The nodes are then mapped to a low-dimensional space that can be easily segregated to form clusters. Spectral Clustering uses information from the eigenvalues (spectrum) of special matrices (i.e. Affinity Matrix, Degree Matrix and Laplacian Matrix) derived from the graph or the data set.

Differences between Spectral Clustering and Conventional Clustering Techniques

Spectral clustering is flexible and allows us to cluster non-graphical data as well. It makes no assumptions about the form of the clusters. Clustering techniques, like **K-Means, assume that the points assigned to a cluster are spherical about the cluster centre.** This is a strong assumption and may not always be relevant. In such cases, Spectral Clustering helps create more accurate clusters. It can correctly cluster observations that actually belong to the same cluster, but are farther off than observations in other clusters, due to dimension reduction.

The data points in Spectral Clustering should be connected, but may not necessarily have convex boundaries, as opposed to the **conventional clustering techniques**, where clustering **is based on the compactness of data points.** Although, it is **computationally expensive for large datasets,** since eigenvalues and eigenvectors need to be computed and clustering is performed on these vectors. Also, for large datasets, the complexity increases and accuracy decreases significantly.





Basic Terminologies Used in Spectral Clustering

Adjacency and Affinity Matrix (A)

The graph (or set of data points) can be represented as an Adjacency Matrix, where the row and column indices represent the nodes, and the entries represent the absence or presence of an edge between the nodes (i.e. if the entry in row 0 and column 1 is 1, it would indicate that node 0 is connected to node 1). An Affinity Matrix is like an Adjacency Matrix, except the value for a pair of points expresses how similar those points are to each other. If pairs of points are very dissimilar then the affinity should be 0. If the points are identical, then the affinity might be 1. In this way, the affinity acts like the weights for the edges on our graph.

Degree Matrix (D)

A Degree Matrix is a diagonal matrix, where the degree of a node (i.e. values) of the diagonal is given by the number of edges connected to it. We can also obtain the degree of the nodes by taking the sum of each row in the adjacency matrix.

Laplacian Matrix (L)

This is another representation of the graph/data points, which attributes to the beautiful properties leveraged by Spectral Clustering. One such representation is obtained by subtracting the Adjacency Matrix from the Degree Matrix **(i.e. L = D – A).** To gain insights and perform clustering, the eigenvalues of L are used. Some useful ones are mentioned below:

Spectral Gap

The first non-zero eigenvalue is called the Spectral Gap. The Spectral Gap gives us some notion of the density of the graph.

Fiedler Value

The second eigenvalue is called the Fiedler Value, and the corresponding vector is the Fiedler vector. Each value in the Fiedler vector gives us information as to which side of the decision boundary a particular node belongs to.

Laplacian Matrix

Using **L**, we find the first large gap between eigen values which generally indicates that the number of eigenvalues before this gap is equal to the number of clusters.

'Cluster Eigenspace Problem' Simplified

To find optimal clusters, the Laplacian matrix (L) should approximately be a block diagonal matrix, where each block represents a cluster. Each of these blocks consists of sub-blocks, that help us identify clusters with non-convex boundaries. The requirement of Spectral Clustering to find the actual cluster labels is such that the lowest eigenvalue and eigenvector pairs in the full space of the Laplacian Matrix should belong to different clusters. This happens when these eigenvectors correspond to the lowest eigenvectors in one of these sub-blocks present in the sub-spaces of the Laplacian Matrix. This restricts the eigenvalue spectrum of L, so that the set of lowest full-space eigenvalues consist of the lowest sub-block eigenvalues, which otherwise could have given us more meaningful insights of the graph/data points.

Emphasizing the Constraints on Real World Data

The constraint on the eigenvalue spectrum suggests that **Spectral Clustering will only work on fairly uniform datasets;** N uniformly sized clusters. Otherwise, there is no guarantee that the full space and sub-block eigen spectrums will line up nicely. In general, Spectral Clustering can be **quite sensitive to changes in the similarity graph** and to its parameters. Additionally, the selection of the similarity graph (i.e. K-Nearest Neighbours graph and e-neighbourhood graph) can affect results, as they respond differently to the data present in different densities and scales.

Furthermore, there can be **consistency issues** while using unnormalized Spectral Clustering algorithms, caused by the possible failure to converge the data points to suitable clusters as more data points are added. The **A** that is used to obtain the **L** is essential in determining similarity within data points, and hence, if **A** is not defined properly, poor similarity measures deter the applicability of Spectral Clustering. Unfortunately, there has been no systematic study which investigates the effects of the similarity graph and its parameters on clustering. Therefore, there are no well-justified rules of thumb.

Implementation in Python

Following are the files which have Implementation of Spectral Clustering in Python

References for further reading:

- Spectral Clustering: A Quick Overview
- Spectral Clustering - Foundation and Application
- Spectral Clustering from Scratch
- A Tutorial on Spectral Clustering

-Authored by
[Mayank Lal,](#)
Analyst at Absolutdata

Unsupervised Learning (like Clustering) Using Random Forest

Folk-Wisdom's Fallacy

A New Approach

Machine learning methods are often categorized as supervised (outcome labels are used) or unsupervised (outcome labels are not used).

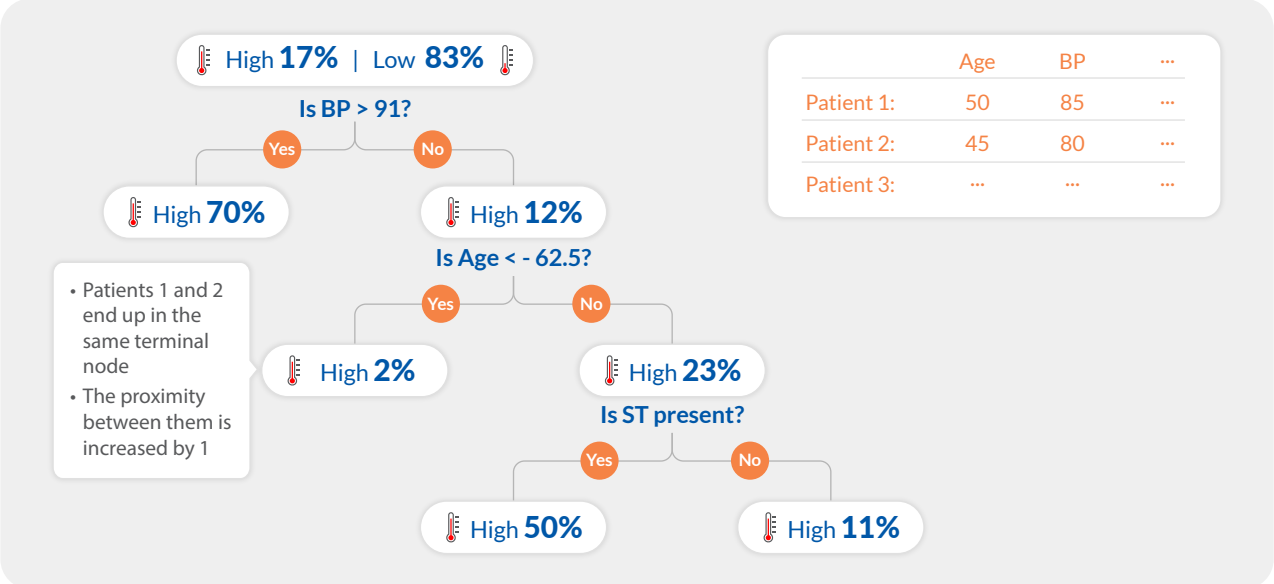
Most of us are of the opinion that techniques like Random Forest, SVM, Logistic, etc. can only be used for supervised learning. However, many supervised methods can be turned into unsupervised methods using the following procedure:

An artificial class label is created that distinguishes the 'observed' data from suitably generated 'synthetic' data. The observed data is the original unlabeled data, while the synthetic data is drawn from a reference distribution. Supervised learning methods, which distinguish observed data from synthetic data, yield a dissimilarity measure that can be used as input in subsequent unsupervised learning methods.

As stated above, many unsupervised learning methods require the inclusion of an input dissimilarity measure among the observations. Hence, if a dissimilarity matrix can be produced using Random Forest, we can successfully implement unsupervised learning. The patterns found in the process will be used to make clusters.

How Do We Generate a Dissimilarity Matrix?

- Terminal tree nodes contain few observations. If case 'i' and case 'j' both land in the same terminal node, we increase the similarity between 'i' and 'j' by 1.
- At the end of the run, divide by 2 x no. of trees.
- Dissimilarity = sqrt(1-Similarity).



Steps for Random Forest Clustering

1. Label the observed data as class 1.
2. Generate synthetic observations and label them as class 2.
    - There are two standard ways of generating synthetic observations:
        - Independent sampling from each of the univariate distributions of the variables (Addcl1 =independent marginals).
        - Independent sampling from uniforms, such that each uniform has a range equal to the range of the corresponding variable (Addcl2).
3. Construct an RF predictor to distinguish class 1 from class 2.
4. Use the resulting dissimilarity measure in unsupervised analysis.
5. Compute distance matrix from RF: distance matrix = sqrt(1-similarity matrix).
6. Conduct partitioning around medoid (PAM) clustering analysis where the input parameter = no. of clusters k.

Random Forest Clustering in Research

RF dissimilarity has been successfully used in several unsupervised learning tasks involving genomic data:

1. Breiman and Cutler (2003) applied RF clustering to DNA microarray data.
2. Allen et al. (2003) applied it to genomic sequence data.
3. Shi et al. (2004) applied it to tumor-marker data.

In these real data applications, the resulting clusters often made sense in their biology applications, which provides indirect empirical evidence that this method works well in practice.

Implementing Random Forest Clustering in R

References

Full Manuscript RFclustering

Rfclustering

Clustering Through Decision Tree Construction

-Authored by

Himanshu Keshav,
Data Scientist at Absolutdata

Clustering Validation Techniques

Experience Extended

Abstract

Clustering is an unsupervised process in data mining and pattern recognition, and most clustering algorithms are very sensitive to their input parameters. Therefore, it is imperative to evaluate the algorithms' outcome. Ideally, the resulting clusters should have good statistical properties (compact, well-separated, connected, and stable) and provide practically relevant results.

It is difficult to define when a clustering result is acceptable, so several clustering validity techniques and indices have been developed.

Conventional Practices

Analysts have conventionally used statistics like F-values and silhouette coefficients (and others) for non-hierarchical or semi-hierarchical clustering solutions; they execute relevant classification exercises on the new cluster membership to figure out if the clusters are intrinsically homogenous and extrinsically heterogeneous. But these metrics provide very limited information about cluster diagnosis. To generalize cluster solutions, the consideration of several important metrics is advisable.

Validation Measures

The following are the most commonly used validity indices:

External Measures	Internal Measures	Stability Measures
Rand Statistic, Jaccard Coefficient, Hubert's Statistic, Normalized Statistic, Fowlkes-Mallows Index, etc.	Connectivity, Dunn Index, etc.	Average Proportion of Non-Overlap (APN), Average Distance (AD), Average Distance between Means (ADM), Figure of Merit (FOM), etc.

There are many other measures that help validate cluster solutions, including the C-Index, the Cubic Clustering Criterion (CCC), the Dindex, the SDindex, the Point-BiserialIndex, and the Calinski-Harabasz (CH), Duda, Pseudo t2, Gamma, Beale, Gplus, Davies-Bouldin, Frey, Hartigan, Tau, The Ratkowsky-Lance, Scott, Marriot, Ball, Trcovw, Tracew, Friedman, McClain-Rao, Rubin, KL, Gap, and SDbw indices.

Click here for the jupyter notebook with the implementation of some of these measures in python.

On Clustering Validation Techniques

Python Implementation for Cluster Validation

-Authored by
Shivli Gupta,
Data Scientist at Absolutdata

Clustering Illusion: Clusters Lie in the Eyes of the Beholder!

Food for Thought Experiment

Definition

The **Clustering Illusion** is the tendency to erroneously perceive small samples from random distributions to have significant 'streaks' or 'clusters'. It is caused by the human tendency to under-predict the amount of variability likely to appear (due to chance) in a small sample of random or semi-random data. For instance, if you were to flip a coin and have heads turn up ten times in a row, you might think that the coin is biased. But if you were to flip that coin 1,000 times, the odds of getting ten or more heads in a row are a surprising 62 percent.

Consequences

The Clustering Illusion creates traps for marketers. If they figure out some meaningful patterns in a random jumble of information, they tend to wrongly generalize the same patterns onto a larger dataset. A winning streak may indicate the clustering exercise is sound, but it may also be a statistical anomaly.

Is More Data a Quick Fix?

It is true that many things experience short, intense cycles. But it's also true that things tend to regress to the mean over longer periods. So, if an investment for example has a long-term record of good returns but goes through a period of lower returns, it doesn't mean this investment will give a poor return going forward. In fact, it might mean just the opposite: it might make it an even better time to invest. In business and economics, you often have to look at 10-, 15-, or 20-year periods to see the real trends.

Getting more data is helpful, but only if it is drawn at random. If the additional data is fraught with biases induced by something like stratified sampling, then the Clustering Illusion will persist. There will be a risk of generalization of clustering results brought on by the lack of statistical validity in the clustering outputs.

How to Minimize Clustering Illusions

1. Don't place too much emphasis on short-term performance, whether positive or negative. Remember, hot and cold streaks are common and can be due to nothing more than luck.

2 While clustering, don't rely on your intuition to guide you. Instead, use fact-based rules and disciplined strategies to overcome cognitive biases like the cluster illusion.

3 The more often you look at them, the more likely you will see trends that don't really exist.

Food for Thought

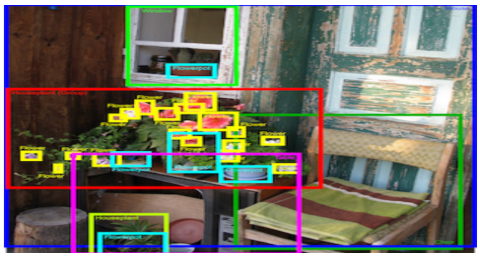
Clustering Illusion demonstrates how hardheaded us mere mortals can be when it comes to facing facts that don't support our beliefs. Awareness of Clustering Illusion can help avoid the trap of patterns that don't exist, and can improve the accuracy and consistency of your clustering outputs.

-Authored by
Bipin Kapri,
Data Scientist at Absolutdata

Data Science Competitions

Active Competitions (as on 28-Aug-2019)

Open Images 2019 - Object Detection



Host : Kaggle
Prize Money : $25K

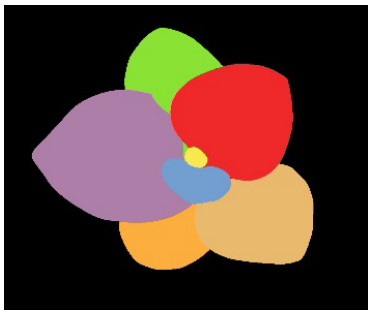
Starts at: 24-Sep-19
Closes on: 01-Oct-19

Open Images 2019 - Instance Segmentation

Host : Kaggle
Prize Money : $25K

Starts at: 24-Sep-19
Closes on: 01-Oct-19

Leaf Segmentation Challenge (LSC)



Host : Codalab
Prize Money : NA

Starts at: 22-Sep-19
Closes on: Never

For more details, please visit the following link:
https://www.kaggle.com/competitions
https://competitions.codalab.org/competitions/

Thank You

For reading this edition of BrainWave from the Absolutdata Labs Team. This digest focuses on some technical angles of analytics and data science. BrainWave is published about 4 times a year. If you haven't already, please subscribe so you receive future editions.

Subscribe