Analytics Vidhya
Learn everything about analytics

An Introduction to Clustering and different methods of clustering

BUSINESS ANALYTICS MACHINE LEARNING

SAURAV KAUSHIK , NOVEMBER 3, 2016 / 24 SHARE

Introduction

Have you come across a situation when a Chief Marketing Officer of a company tells you – "Help me understand our customers better so that we can market our products to them in a better manner!"

I did and the analyst in me was completely clueless what to do! I was used to getting specific problems, where there is an outcome to be predicted for various set of conditions. But I had no clue, what to do in this case. If the person would have asked me to calculate Life Time Value (LTV) or propensity of Cross-sell, I wouldn't have blinked. But this question looked very broad to me!

This is usually the first reaction when you come across a unsupervised learning problem for the first time! You are not looking for specific insights for a phenomena, but what you are looking for are structures with in data with out them being tied down to a specific outcome.

The method of identifying similar groups of data in a data set is called clustering. Entities in each group are comparatively more similar to entities of that group than those of the other groups. In this article, I will be taking you through the types of clustering, different clustering algorithms and a comparison between two of the most commonly used cluster methods.

Let's get started.

Table of Contents

1. Overview
2. Types of Clustering
3. Types of Clustering Algorithms
4. K Means Clustering
5. Hierarchical Clustering
6. Difference between K Means and Hierarchical clustering
7. Applications of Clustering
8. Improving Supervised Learning algorithms with clustering

1. Overview

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups. In simple words, the aim is to segregate groups with similar traits and assign them into clusters.

Let's understand this with an example. Suppose, you are the head of a rental store and wish to understand preferences of your costumers to scale up your business. Is it possible for you to look at details of each costumer and devise a unique business strategy for each one of them? Definitely not. But, what you can do is to cluster all of your costumers into say 10 groups based on their purchasing habits and use a separate strategy for costumers in each of these 10 groups. And this is what we call clustering.

Now, that we understand what is clustering. Let's take a look at the types of clustering.

2. Types of Clustering

Broadly speaking, clustering can be divided into two subgroups :

- **Hard Clustering:** In hard clustering, each data point either belongs to a cluster completely or not. For example, in the above example each customer is put into one group out of the 10 groups.
- **Soft Clustering**: In soft clustering, instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned. For example, from the above scenario each costumer is assigned a probability to be in either of 10 clusters of the retail store.

3. Types of clustering algorithms

Since the task of clustering is subjective, the means that can be used for achieving this goal are plenty. Every methodology follows a different set of rules for defining the '*similarity*' among data points. In fact, there are more than 100 clustering algorithms known. But few of the algorithms are used popularly, let's look at them in detail:

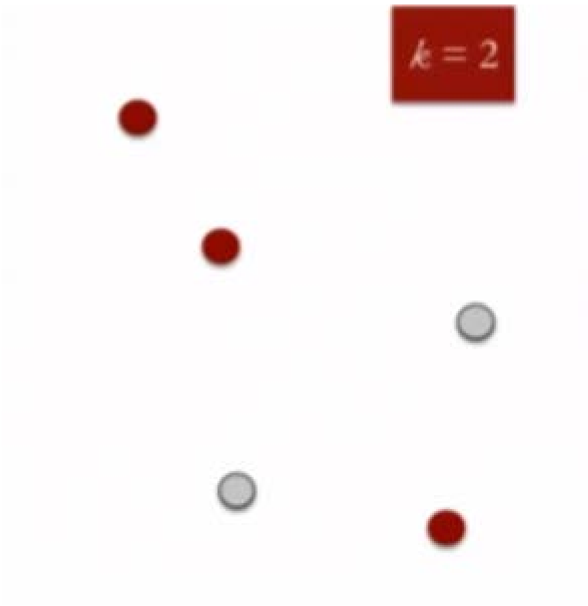
- **Connectivity models:** As the name suggests, these models are based on the notion that the data points closer in data space exhibit more similarity to each other than the data points lying farther away. These models can follow two approaches. In the first approach, they start with classifying all data points into separate clusters & then aggregating them as the distance decreases. In the second approach, all data points are classified as a single cluster and then partitioned as the distance increases. Also, the choice of distance function is subjective. These models are very easy to interpret but lacks scalability for handling big datasets. Examples of these models are hierarchical clustering algorithm and its variants.
- **Centroid models:** These are iterative clustering algorithms in which the notion of similarity is derived by the closeness of a data point to the centroid of the clusters. K-Means clustering algorithm is a popular algorithm that falls into this category. In these models, the no. of clusters required at the end have to be mentioned beforehand, which makes it important to have prior knowledge of the dataset. These models run iteratively to find the local optima.
- **Distribution models:** These clustering models are based on the notion of how probable is it that all data points in the cluster belong to the same distribution (For example: Normal, Gaussian). These models often suffer from overfitting. A popular example of these models is Expectation-maximization algorithm which uses multivariate normal distributions.
- **Density Models:** These models search the data space for areas of varied density of data points in the data space. It isolates various different density regions and assign the data points within these regions in the same cluster. Popular examples of density models are DBSCAN and OPTICS.

Now I will be taking you through two of the most popular clustering algorithms in detail – K Means clustering and Hierarchical clustering. Let's begin.

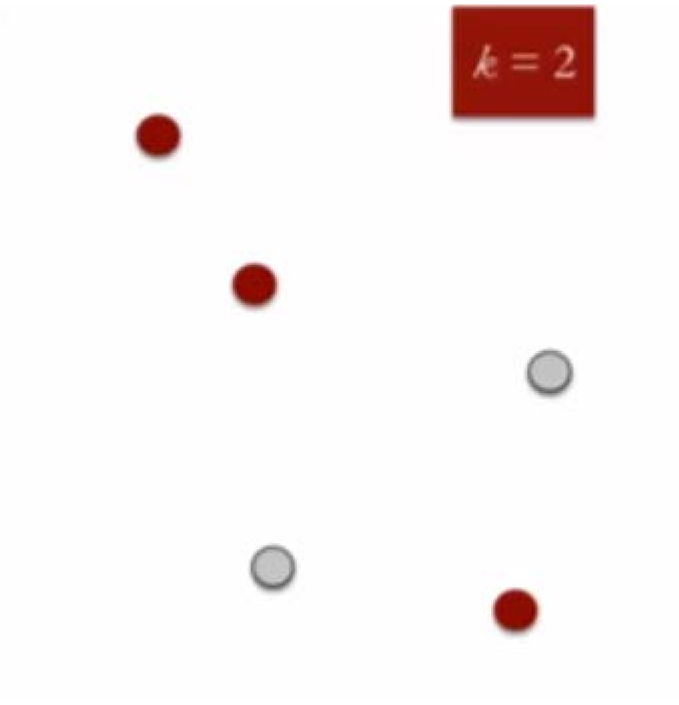
4. K Means Clustering

K means is an iterative clustering algorithm that aims to find local maxima in each iteration. This algorithm works in these 5 steps :

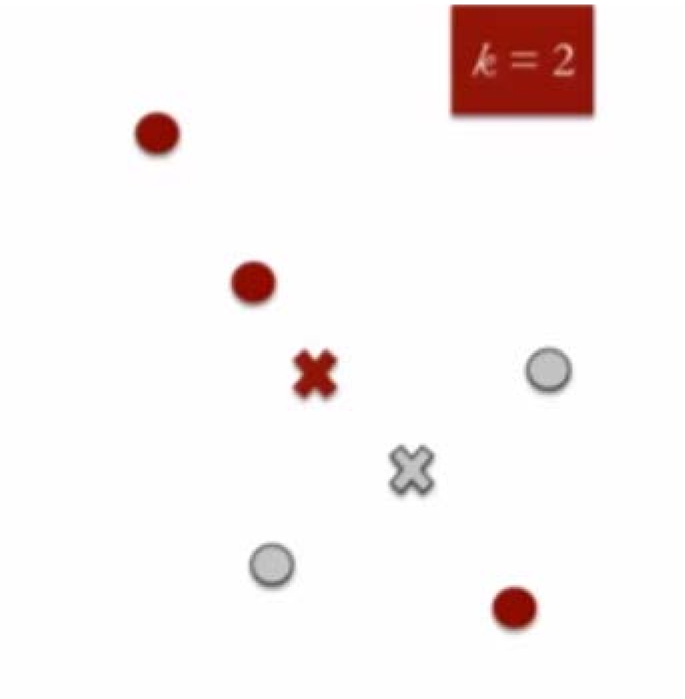
1. Specify the desired number of clusters K : Let us choose k=2 for these 5 data points in 2-D space.



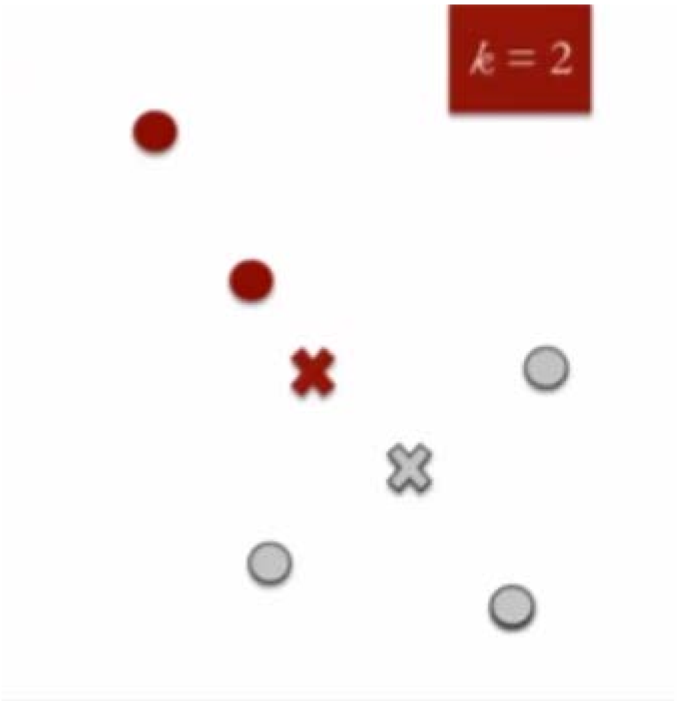
2. Randomly assign each data point to a cluster : Let’s assign three points in cluster 1 shown using red color and two points in cluster 2 shown using grey color.



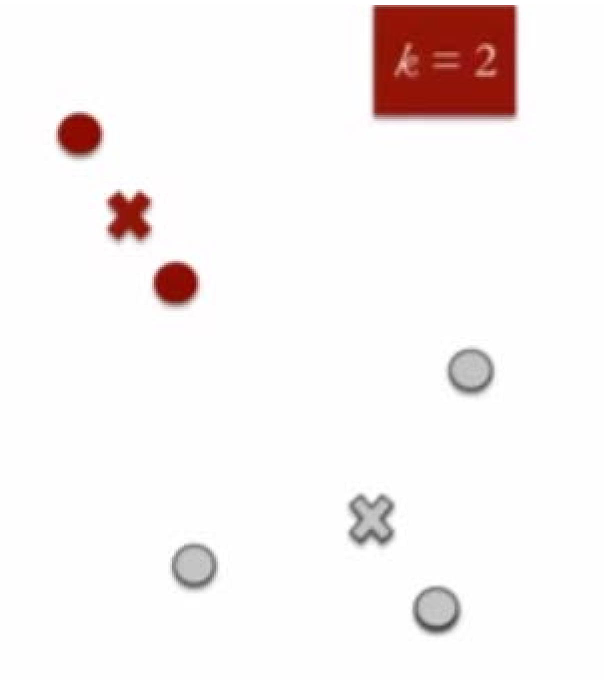
3. Compute cluster centroids : The centroid of data points in the red cluster is shown using red cross and those in grey cluster using grey cross.



4. Re-assign each point to the closest cluster centroid : Note that only the data point at the bottom is assigned to the red cluster even though its closer to the centroid of grey cluster. Thus, we assign that data point into grey cluster



5. Re-compute cluster centroids : Now, re-computing the centroids for both the clusters.

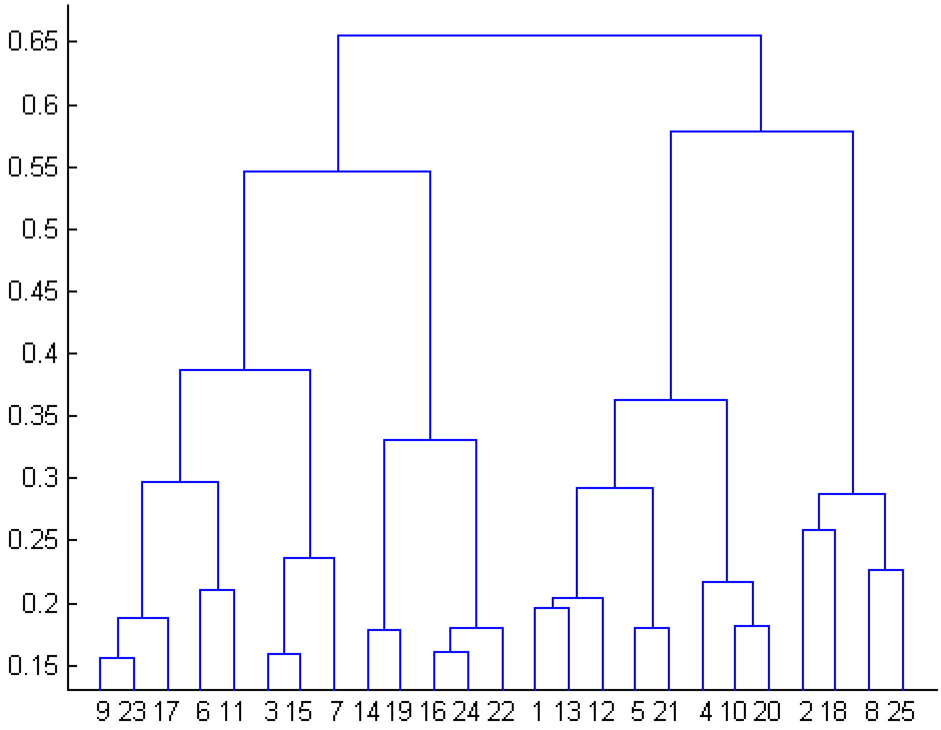


6. Repeat steps 4 and 5 until no improvements are possible : Similarly, we'll repeat the 4th and 5th steps until we'll reach global optima. When there will be no further switching of data points between two clusters for two successive repeats. It will mark the termination of the algorithm if not explicitly mentioned.

5. Hierarchical Clustering

Hierarchical clustering, as the name suggests is an algorithm that builds hierarchy of clusters. This algorithm starts with all the data points assigned to a cluster of their own. Then two nearest clusters are merged into the same cluster. In the end, this algorithm terminates when there is only a single cluster left.

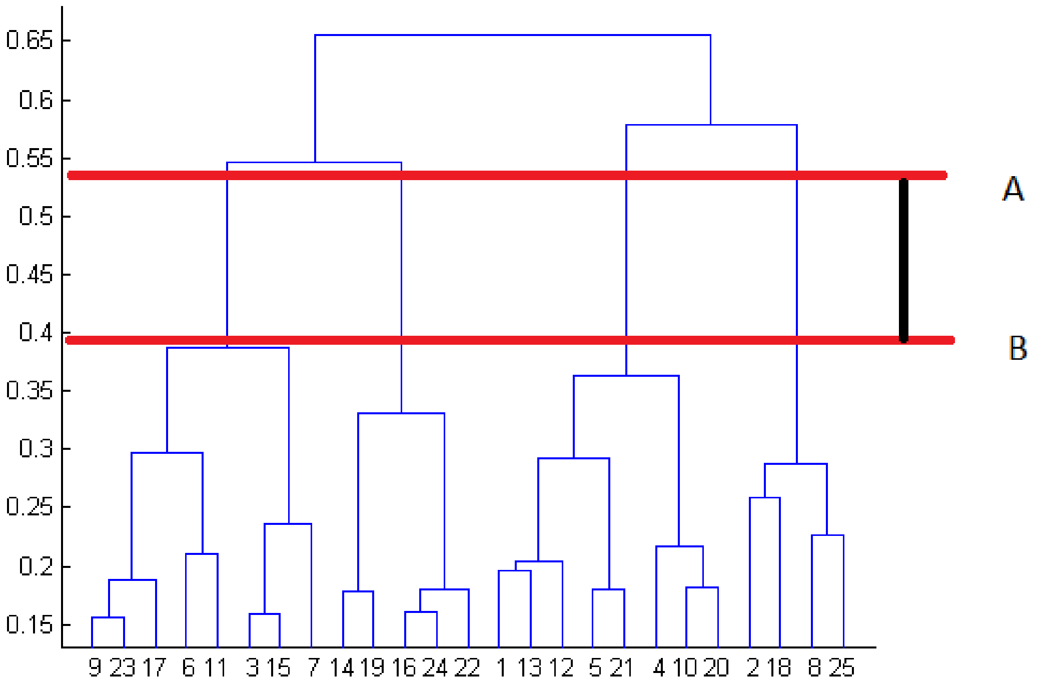
The results of hierarchical clustering can be shown using dendrogram. The dendrogram can be interpreted as:



At the bottom, we start with 25 data points, each assigned to separate clusters. Two closest clusters are then merged till we have just one cluster at the top. The height in the dendrogram at which two clusters are merged represents the distance between two clusters in the data space.

The decision of the no. of clusters that can best depict different groups can be chosen by observing the dendrogram. The best choice of the no. of clusters is the no. of vertical lines in the dendrogram cut by a horizontal line that can transverse the maximum distance vertically without intersecting a cluster.

In the above example, the best choice of no. of clusters will be 4 as the red horizontal line in the dendrogram below covers maximum vertical distance AB.



Two important things that you should know about hierarchical clustering are:

- This algorithm has been implemented above using bottom up approach. It is also possible to follow top-down approach starting with all data points assigned in the same cluster and recursively performing splits till each data point is assigned a separate cluster.
- The decision of merging two clusters is taken on the basis of closeness of these clusters. There are multiple metrics for deciding the closeness of two clusters :
  - Euclidean distance: $||a-b||_2 = \surd(\Sigma(a_i-b_i))$
  - Squared Euclidean distance: $||a-b||_2^2 = \Sigma((a_i-b_i)^2)$
  - Manhattan distance: $||a-b||_1 = \Sigma|a_i-b_i|$
  - Maximum distance:$||a-b||_{INFINITY} = \max_i|a_i-b_i|$
  - Mahalanobis distance: $\surd((a-b)^T S^{-1} (-b))$ {where, s : covariance matrix}

6. Difference between K Means and Hierarchical clustering

- Hierarchical clustering can’t handle big data well but K Means clustering can. This is because the time complexity of K Means is linear i.e. O(n) while that of hierarchical clustering is quadratic i.e. $O(n^2)$.
- In K Means clustering, since we start with random choice of clusters, the results produced by running the algorithm multiple times might differ. While results are reproducible in Hierarchical clustering.
- K Means is found to work well when the shape of the clusters is hyper spherical (like circle in 2D, sphere in 3D).
- K Means clustering requires prior knowledge of K i.e. no. of clusters you want to divide your data into.

But, you can stop at whatever number of clusters you find appropriate in hierarchical clustering by interpreting the dendrogram

7. Applications of Clustering

Clustering has a large no. of applications spread across various domains. Some of the most popular applications of clustering are:

- Recommendation engines
- Market segmentation
- Social network analysis
- Search result grouping
- Medical imaging
- Image segmentation
- Anomaly detection

8. Improving Supervised Learning Algorithms with Clustering

Clustering is an unsupervised machine learning approach, but can it be used to improve the accuracy of supervised machine learning algorithms as well by clustering the data points into similar groups and using these cluster labels as independent variables in the supervised machine learning algorithm? Let's find out.

Let's check out the impact of clustering on the accuracy of our model for the classification problem using 3000 observations with 100 predictors of stock data to predicting whether the stock will go up or down using R. This dataset contains 100 independent variables from X1 to X100 representing profile of a stock and one outcome variable Y with two levels : 1 for rise in stock price and -1 for drop in stock price.

The dataset is available here : Download

Let's first try applying randomforest without clustering.

```
#loading required libraries
```

```
library('randomForest')


library('Metrics')
```

```
#set random seed
set.seed(101)

#loading dataset

data<-read.csv("train.csv",stringsAsFactors= T)

#checking dimensions of data
dim(data)

## [1] 3000  101

#specifying outcome variable as factor


 data$Y<-as.factor(data$Y)

#dividing the dataset into train and test
train<-data[1:2000,]
test<-data[2001:3000,]

#applying randomForest
model_rf<-randomForest(Y~.,data=train)

preds<-predict(object=model_rf,test[,-101])

table(preds)

## preds
##  -1   1
## 453 547

#checking accuracy

auc(preds,test$Y)

## [1] 0.4522703
```

So, the accuracy we get is 0.45. Now let’s create five clusters based on values of independent variables using k-means clustering and reapply randomforest.

```
#combing test and train


all<-rbind(train,test)


#creating 5 clusters using K- means clustering


Cluster <- kmeans(all[,-101], 5)


#adding clusters as independent variable to the dataset.
all$cluster<-as.factor(Cluster$cluster)


#dividing the dataset into train and test
train<-all[1:2000,]
test<-all[2001:3000,]


#applying randomforest
model_rf<-randomForest(Y~.,data=train)


preds2<-predict(object=model_rf,test[,-101])


table(preds2)


## preds2


## -1   1


##548 452


auc(preds2,test$Y)


## [1] 0.5345908
```

Whoo! In the above example, even though the final accuracy is poor but clustering has given our model a significant boost from accuracy of 0.45 to slightly above 0.53.

This shows that clustering can indeed be helpful for supervised machine learning tasks.

End Notes

In this article, we have discussed what are the various ways of performing clustering. It find applications for unsupervised learning in a large no. of domains. You also saw how you can improve the accuracy of your supervised machine learning algorithm using clustering.

Although clustering is easy to implement, you need to take care of some important aspects like treating outliers in your data and making sure each cluster has sufficient population. These aspects of clustering are dealt in great detail in this article.

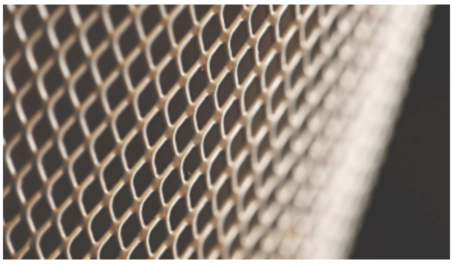
Did you enjoyed reading this article? Do share your views in the comment section below.

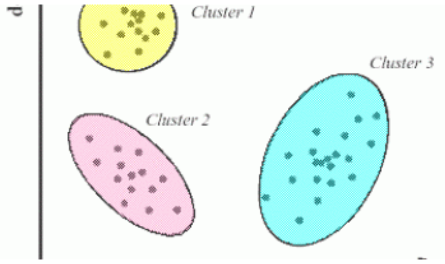
Got expertise in Machine Learning / Big Data / Data Science? Show case your knowledge and help Analytics Vidhya community by posting your blog.

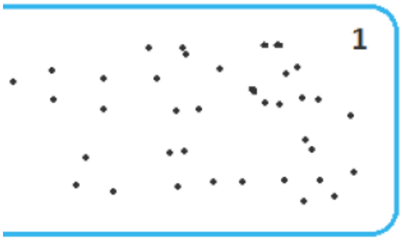
Share this:

□ 667 □ 430 □ □ □ □

RELATED







40 Questions to test a Data Scientist on Clustering Techniques (Skill test Solution)
February 5, 2017
In "Business Analytics"

Getting your clustering right (Part I)
November 12, 2013
In "Business Analytics"

Getting your clustering right (Part II)
November 17, 2013
In "Big data"

TAGS: , , , , ,
,

Next Article

Creating an artificial artist: Color your photos using Neural Networks

Previous Article

Investigation on handling Structured & Imbalanced Datasets with Deep Learning

Author

Saurav Kaushik

Saurav is a Data Science enthusiast, currently in the final year of his graduation at MAIT, New Delhi. He loves to use machine learning and analytics to solve complex data problems.

This is article is quiet old now and you might not get a prompt response from the author. We would request you to post this comment on Analytics Vidhya **Discussion portal** to get your queries resolved.

24 COMMENTS

Ankit Gupta says:

REPLY

NOVEMBER 3, 2016 AT 8:54 AM

Very nice tutorial Saurav!

Saurav Kaushik says:

REPLY

NOVEMBER 3, 2016 AT 10:43 AM

Good to see you liked it. Thank you!

Richard Warnung says:

REPLY

NOVEMBER 3, 2016 AT 9:40 AM

Nice, post! Please correc the last link – it is broken – thanks!

Saurav Kaushik says:

REPLY

NOVEMBER 3, 2016 AT 10:41 AM

Hi, Richard.

Glad you liked it !

Thanks for pointing out. Its fixed now!

Sai Satheesh G says:

REPLY

NOVEMBER 3, 2016 AT 11:45 AM

I accept that clustering may help in improving the supervised models. But here in the above:

Clustering is performed on sample points (4361 rows). Is that right.?

But I think correct way is to cluster features (X1-X100) and to represent data using cluster representatives and then perform supervised learning.

Can you please elaborate further? Why samples are being clustered in the code (not independent variables)?

Saurav Kaushik says:

REPLY

NOVEMBER 3, 2016 AT 3:03 PM

Hey, Sai.

So, Yes. I have clustered the observations (or rows, 3000 in total).

Consider all these data points (observations) in data space with all the features (x1-x100) as dimensions.
What I'm doing is to cluster these data points into 5 groups and store the cluster label as a new feature itself.

Clustering the 100 independent variables will give you 5 groups of independent variables. Going this way, how exactly do you plan to use these cluster labels for supervised learning?

sai satheesh says:

REPLY

NOVEMBER 3, 2016 AT 3:55 PM

1. I am not able to understand (intuitively) why clustering sample points will yield better results? Please explain and if you have any book/paper explaining this , please provide it too.

2. Regarding what I said , I read about this PAM clustering method (somewhat similar to k-means) , where one can select representative objects (represent cluster using this feature, for example if X1-X10 are in one cluster , may be one can pick X6 to represent the cluster , this X6 is provided by PAM method). Then classification is performed simply on those objects. I am not sure whether that would yield better results. Just wanted to share this.

I guess this dataset is from a hackathlon , even I worked on that problem. If you did too, what method you chose for clustering ?

Amit says:

REPLY

NOVEMBER 6, 2016 AT 6:02 PM

Once you have separated the data into 5 clusters, can we create five different models for the 5 clusters. What are your thoughts?

Nice introductory article by the way. In the next article, may be you can discuss about identifying clusterability of the data, finding the ideal number of clusters for the k-Means. Also how can we evaluate our clustering model?

Luis says:

REPLY

NOVEMBER 3, 2016 AT 2:31 PM

Nice article!
How would you handle a clustering problem when there are some variables with many missing values (let’s say…around 90% of each column). These missing values are not random at all, but even they have a meaning, the clustering output yields some isolated (and very small) groups due to these missing values.
Thanks in advance!

Saurav Kaushik says:

REPLY

NOVEMBER 3, 2016 AT 3:32 PM

Hi Luis. Thanks.

1. Since the missing values are as high as 90%, you can consider dropping these variables.
2. As you said, these missing values are not completely meaningless, try imputing them (might not yield good results with this high percentage of missing values.)
3. If the pattern in missing values is something like say… values are missing because students didn't took a certain test otherwise that column contains the scores of that test. You can try replacing the variable with another variable having 0 for missing values and 1 for some valid value.

Luis says:

REPLY

NOVEMBER 4, 2016 AT 10:46 AM

Hi again,

Thanks for the response. Let's say I cannot drop these variables, so I have to impute them somehow. What would affect less to a distance function (such as Euclidan), median or mean?

As in your example, there are students that did not take the test, so I do not want them to affect the output. However, students who took the test should be meaningful and It is important whether they got a bad score or a good one.

Saurav Kaushik says:

REPLY

NOVEMBER 5, 2016 AT 6:14 AM

Choice of central tendency depends on your data. Mean is generally a good central tendency to impute your missing values with. But consider a situation in which you have to impute salaries of employees in an organization. The CEO, Directors, etch will have very high salaries but majority will have comparatively very lower salary. So in that case, median should be the way to go.

Ok, so to handle example similar to that, create another column in your data with 0 for rows that have missing values for your column under consideration and 1 for some valid value. And in the main column, replace all NA with some unique value.

Hope this resolves your query.

Kunal Dash says:

REPLY

NOVEMBER 3, 2016 AT 4:05 PM

Hello Saurav,

Your article and related explanation on clustering and the two most used methods was very insightful. However, please do enlighten us by telling how does one interpret cluster output for both these methods – K-means and Hierarchical. Also, it would be nice if you could let the reader know when could one use K-means versus say something like K-median. In what scenario does the former work and in which one does the latter???

It would also be a great idea to:
1. Discuss the ways to implement a density based algorithm and a distribution based one
2. Maybe show an actual example of market segmentation

You have done a good job of showing how clustering could in sense preclude a following classification method but if the problem is such that it is only limited to clustering, then how would you explain the output to an uninitiated audience?

Maybe some thoughts for your second article in the clustering article. But great job. I enjoyed reading your piece.

Saurav Kaushik says:

REPLY

NOVEMBER 5, 2016 AT 6:30 AM

Hi Kunal,

I'm happy that you liked the article. Actually, clustering is a very wide topic to be completely covered in a single article. For some of the things that you mentioned like when to use which method out of two , you can refer to differences between two.

For interpretation of Clusters formed using say Hierarchical clustering is depicted using dendrograms.

Apart from these, things like using density based and distribution based clustering methods, market segmentation could definitely be a part of future articles on clustering.

Thank you for your thoughts.

Kunal Dash says:

REPLY

NOVEMBER 3, 2016 AT 4:09 PM

Also Saurav,

It might be a good idea to suggest which clustering algorithm would be appropriate to use when:

1. All variables are continuous
2. All variables are categorical – many times this could be the case
3. All variables are count – maybe sometimes
4. A mix of continuous and categorical – this could be possibly the most common
5. Similarly a mix of continuous, categorical and count

To be more precise, if I had one or more scenarios above, and was using a distance based method to calculate distances between points, what distance calculation method works where.

Any insights would be great!!

Saurav Kaushik says:

REPLY

NOVEMBER 5, 2016 AT 8:57 AM

So, to understand this, its important to understand how categorical variables behave in clustering.

If the levels of your categorical variables are in sequence like : Very bad, bad, Average, Good, Very Good. You can try encoding labels say with 0,1,2,3 and 4 respectively.

If there is no sequence in levels like : red, green and orange , you can try one hot encoding.

Also, there is no one definite best distance metric to cluster your data. It depends on various factors like the ones you mentioned : type of variables. Also, things like the scales of variables , no. of clusters you want are important while deciding the best distance metric.

Nikunj Agarwal says:

REPLY

NOVEMBER 4, 2016 AT 5:17 AM

Hi Saurav,
Since we are classifying assets in this tutorial, don't you think corelation based distance should give us better results than eucledian distances (which k-means normally uses)?

Saurav Kaushik says:

REPLY

NOVEMBER 5, 2016 AT 9:02 AM

Hi Nikunj,

Intuitively speaking, its definitely worth a shot. Good suggestion.

Kern Paillard says:

REPLY

NOVEMBER 7, 2016 AT 10:49 AM

Hi

It 's a good post on covering a broad topic like Clustering. However, I'm not so convinced about using Clustering for aiding Supervised ML.

For me, Clustering based approaches tend to be more 'exploratory' in nature to understand the inherent data structure, segments et al.

Dimensionality Reduction techniques like PCA are more intuitive approaches (for me) in this case, quite simple because you don't get any dimensionality reduction by doing clustering and vice-versa, yo don't get any groupings out of PCA like techniques.

my distinction of the two,
PCA
is used for dimensionality reduction / feature selection / representation learning e.g. when the feature space contains too many irrelevant or redundant features. The aim is to find the intrinsic dimensionality of the data.

K-means
is a clustering algorithm that returns the natural grouping of data points, based on their similarity.

I'd like to point to the excellent explanation and distinction of the two on Quora :
https://www.quora.com/What-is-the-difference-between-factor-and-cluster-analyses

my question to you
how would you fit / cluster the same groupings (you obtained out of clustering the training set) onto a unseen test set? or would you apply clustering to it again?

typically, you perform PCA on a training set and apply the same loadings on to a new unseen test set and not fit a new PCA to it..

Aditya says:

REPLY

NOVEMBER 13, 2016 AT 1:06 AM

Really nice article Saurav , this helped me understand some of the basic concepts regarding clustering.
I was hoping if you can post similar articles on Fuzzy, DBSCAN, Self Organizing Maps.

Aditya

J Vitor da Silva says:

REPLY

NOVEMBER 14, 2016 AT 12:03 PM

Hi Saurav Kaushik,

I am new to this area, but I am in search of help to understand it deeper.
One of my personal projects involves analysing data for creating a "predictive model" based on some information collected about previous historical data which I have in a spreadsheet (or in .txt file if it is bette).

Could you recommend a simple package (in Python or in Delphi) that can help me do something like this?

My spreadsheet has (for example), 1500 lines which represent historical moments (Test 1, Test2…Test1500).
On the columns, I have the Labels and Values for each of 1000 characteristics I analyse separately at each Test.

What I would like to do with this? To be able to “predict” some 10 ou 20 values for 10 or 20 characteristics for the next Test1501.

Do you think it is possible?

If you are involved in this kind of project, what would it cost me to have your help in building a tool for doing that? I can send you an example file, if you would be interested in helping me.

My direct contact : dixiejoelottolex at gmail dot com

Laurent says:

REPLY

NOVEMBER 17, 2016 AT 10:18 AM

Hi and thank you for your article. Running your example I am running in a series of issues.
The first one being the result of preds<-predict(object=model_rf,test[,-101])

```
head(table(preds))
preds
-0.192066666666667 -0.162533333333333 -0.120533333333333 -0.0829333333333333
-0.0793333333333333
1 1 1 1 1
-0.079
1
```

Then

```
auc(preds,test$Y)
[1] NaN
```

The second exemple with the added cluster produces the same result.
Any idea why my result is so different than yours?

Saurav Kaushik says:

REPLY

DECEMBER 10, 2016 AT 7:55 AM

Hey Laurent,

1. Make sure your outcome variable in categorical and so are your predictions.
2. Make sure you have loaded the Metrics package as auc() is the function defined in that package.

Hope this will resolve your query.

Shirish says:

REPLY

DECEMBER 12, 2016 AT 11:40 AM

Hey Saurav,

Could you please give a code for Python?

Thanks

LEAVE A REPLY

Your email address will not be published.

Name (required)

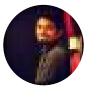
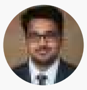
Email (required)

Website

Submit Comment

TOP ANALYTICS VIDHYA USERS

Rank	Name		Points
1		vopani	8204
2		SRK	7707

3		binga	5269
4		Aayushmnit	5258
5		Mark Landry	5243

More Rankings

POPULAR POSTS

- Essentials of Machine Learning Algorithms (with Python and R Codes)
- A Complete Tutorial to Learn Data Science with Python from Scratch
- 15 Trending Data Science GitHub Repositories you can not miss in 2017
- Understanding Support Vector Machine algorithm from examples (along with code)
- A Complete Tutorial on Tree Based Modeling from Scratch (in R & Python)
- Key Highlights in Data Science / Deep Learning / Machine Learning 2017 and What can we Expect in 2018?
- 7 Types of Regression Techniques you should know!
- Fundamentals of Deep Learning – Introduction to Recurrent Neural Networks

RECENT POSTS

Key Highlights in Data Science / Deep Learning / Machine Learning 2017 and What can we Expect in 2018?

FAIZAN SHAIKH , DECEMBER 26, 2017

11 most read Machine Learning articles from Analytics Vidhya in 2017

NSS , DECEMBER 22, 2017

11 most read Deep Learning Articles from Analytics Vidhya in 2017

DISHASHREE GUPTA , DECEMBER 21, 2017

15 Trending Data Science GitHub Repositories you can not miss in 2017

SUNIL RAY , DECEMBER 18, 2017

GET CONNECTED

13,078 FOLLOWERS	41,770 FOLLOWERS
2,410 FOLLOWERS	Email SUBSCRIBE