DATA ANALYTICS

Unit 4: Clustering Algorithms - DBSCAN

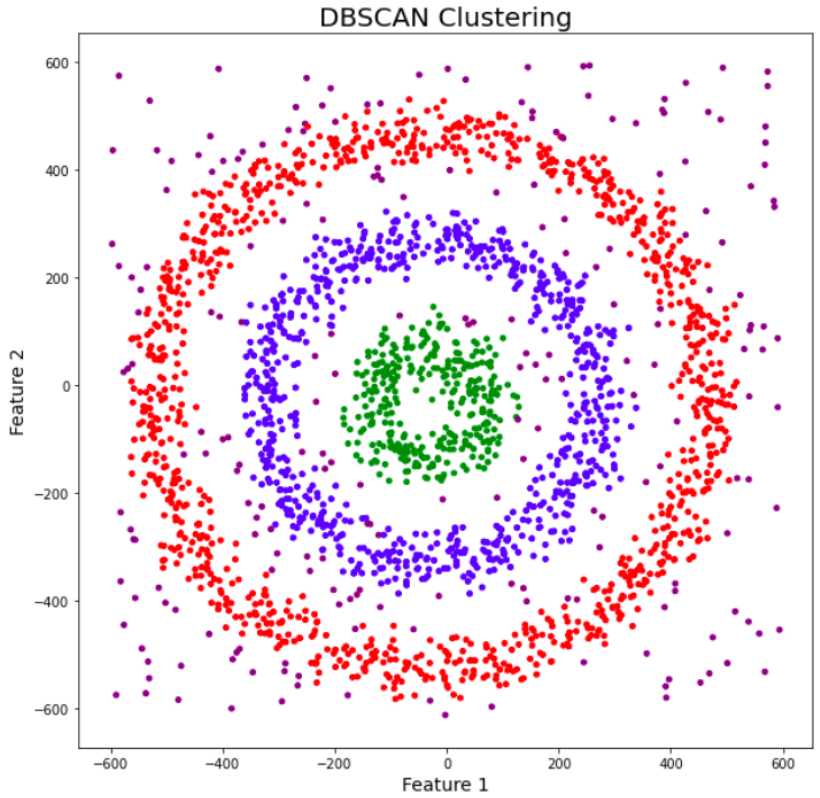
Jyothi R.
Department of Computer Science andEngineering

Why do we need DBSCAN Clustering?

- Three different dense clusters in the form of concentric circles with some noise here.
- Now, let's run K-Means and Hierarchical clustering algorithms and see how they cluster these data points.
- There are four colors in the graph. Noise is considered as a different cluster which is represented by the purple color.
- Sadly, both of them failed to cluster the data points. Also, they were not able to properly detect the noise present in the dataset.

Why do we need DBSCAN Clustering?

- Now, let's take a look at the results from DBSCAN clustering.
- DBSCAN is not just able to cluster the data points correctly, but it also perfectly detects noise in the dataset.



What Exactly is DBSCAN Clustering?

- DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise.
- It was proposed by Martin Ester et al. in 1996. DBSCAN is a density-based clustering algorithm that works on the assumption that clusters are dense regions in space separated by regions of lower density.
- It can identify clusters in large spatial datasets by looking at the local density of the data points.

DBSCAN Clustering

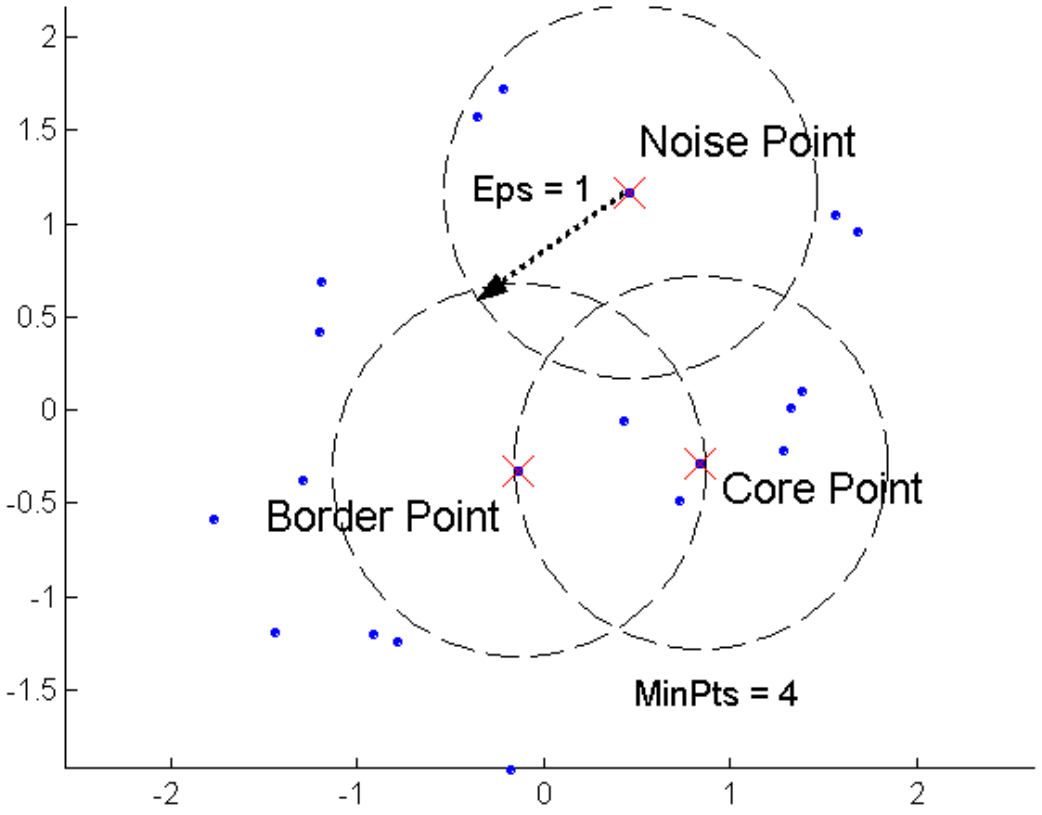
Density = number of points within a specified radius (Eps)

A point is a core point if it has more than a specified number of points (MinPts) within Eps

These are points that are at the interior of a cluster

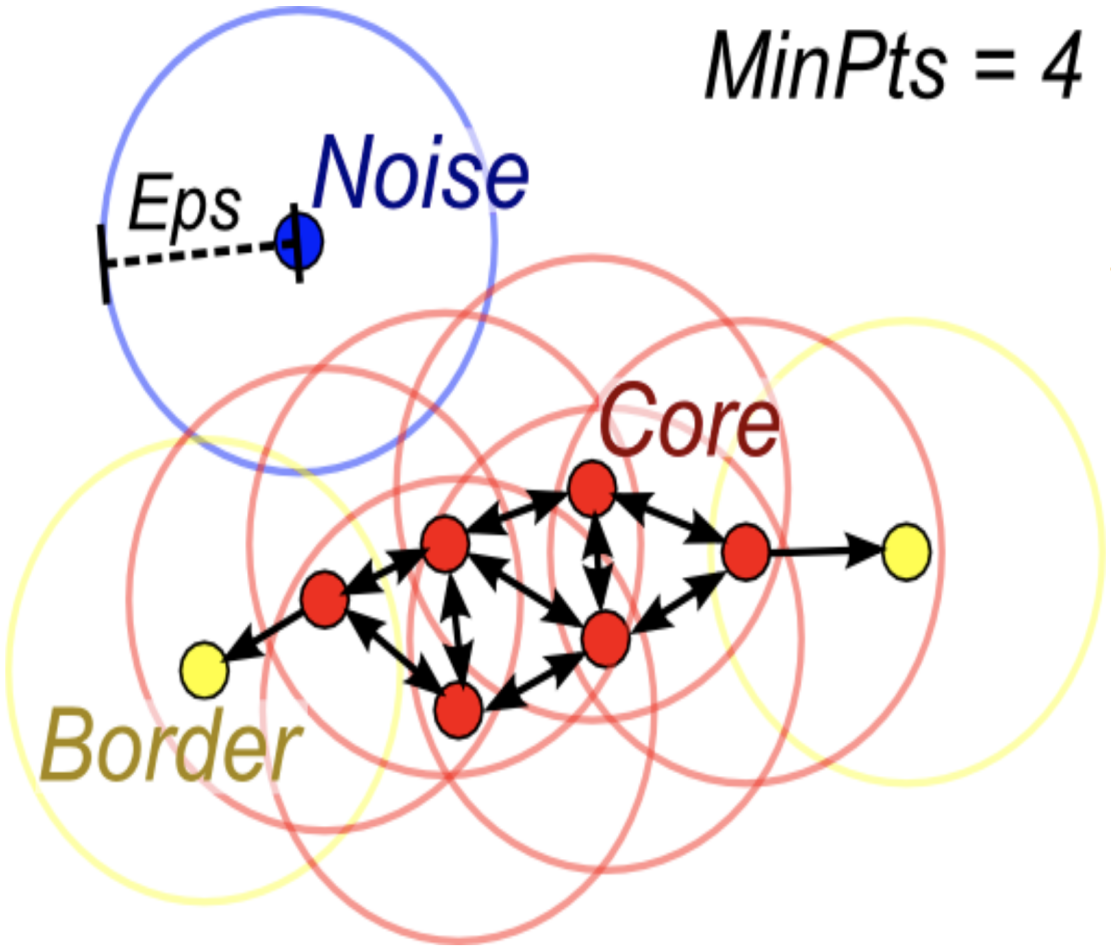
A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point

A noise point is any point that is not a core point or a border point.



Tan, Steinbach and Kumar, "Introduction to Data Mining"

DBSCAN: Core, Border, and Noise Points



Red: Core Points

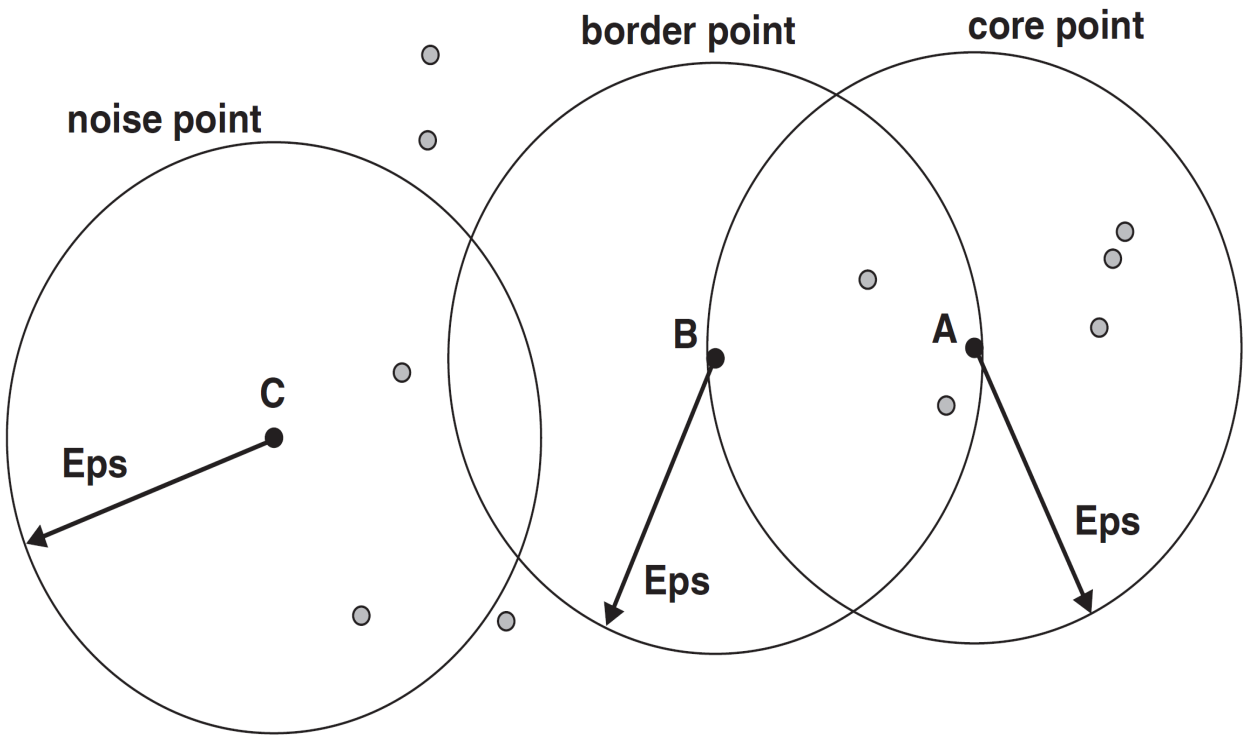
Yellow: Border points. Still part of the cluster because it's within epsilon of a core point, but not does not meet the min_points criteria

Blue: Noise point. Not assigned to a cluster

https://medium.com/@elutins/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818

DBSCAN: Core, Border, and Noise Points

MinPts = 7



DBSCAN Algorithm – Labeling Clusters

- Eliminate noise points
- Perform clustering on the remaining points

```
current_cluster_label ← 1
for all core points do
  if the core point has no cluster label then
    current_cluster_label ← current_cluster_label + 1
    Label the current core point with cluster label current_cluster_label
  end if
  for all points in the Eps-neighborhood, except i^th the point itself do
    if the point does not have a cluster label then
      Label the point with cluster label current_cluster_label
    end if
  end for
end for
```

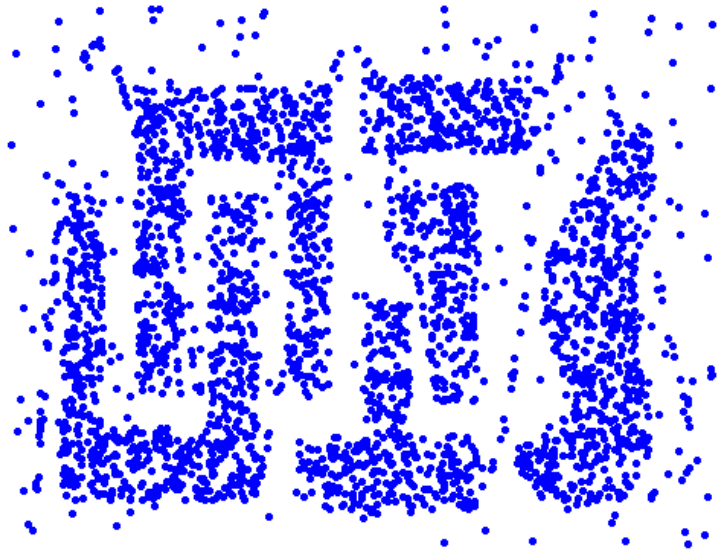
Tan, Steinbach and Kumar, "Introduction to Data Mining"

What Exactly is DBSCAN Clustering?

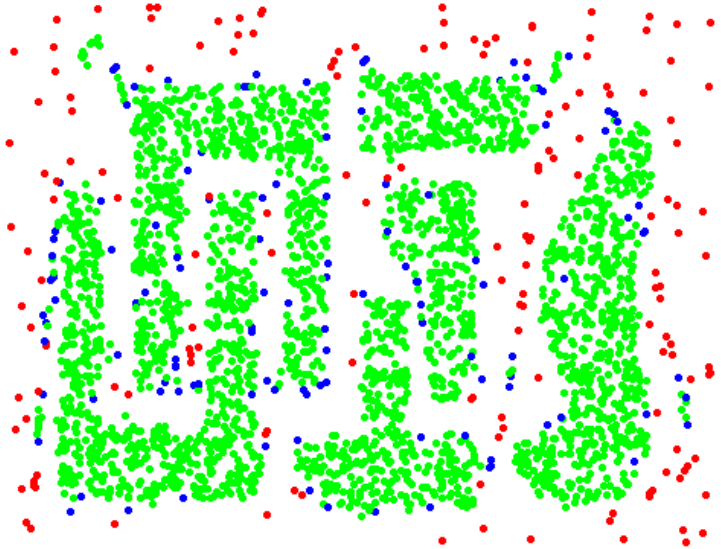
- The most exciting feature of DBSCAN clustering is that it is robust to outliers
- It also does not require the number of clusters to be set beforehand, unlike k-means, where we have to specify the number of centroids
- DBSCAN requires only two parameters: epsilon and minPoints
- Epsilon is the radius of the circle to be created around each data point to check the density

  and
- minPoints is the minimum number of data points required inside that circle for that data point to be classified as a Core point

DBSCAN: Core, Border and Noise Points



Original Points

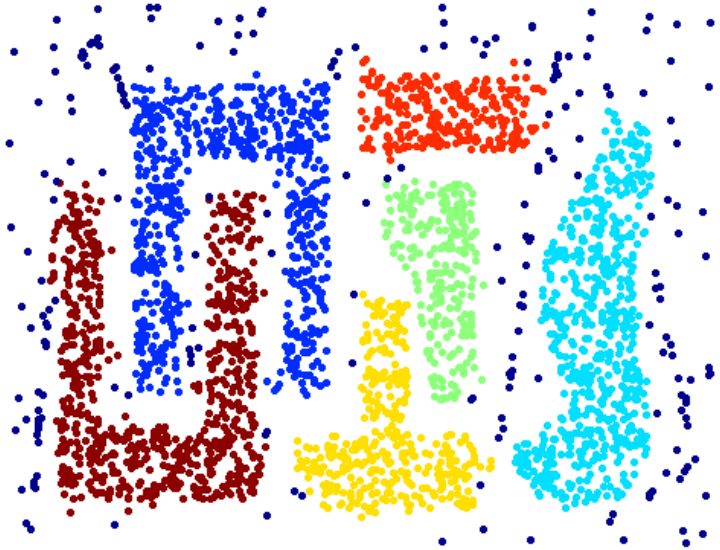


Point types: core, border and noise

Eps = 10, MinPts = 4

When DBSCAN Works Well

Original Points



Clusters

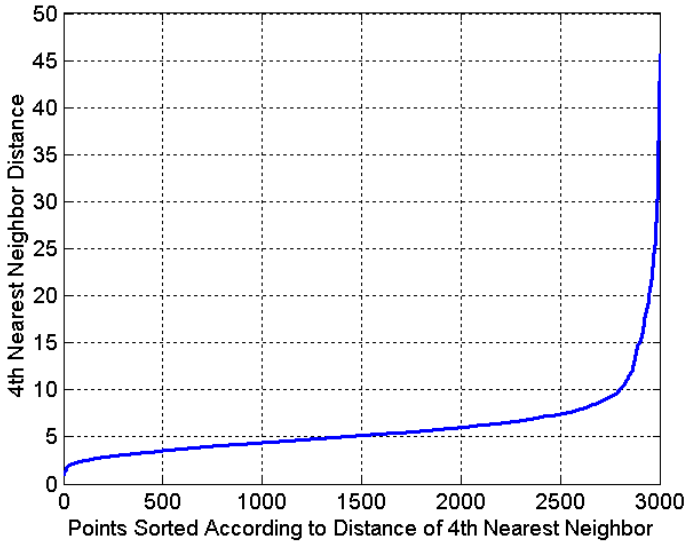
- Resistant to Noise
- Can handle clusters of different shapes and sizes

DBSCAN: Determining EPS and MinPts

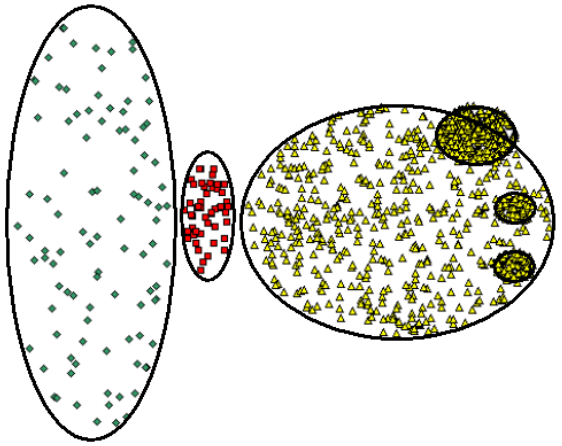
Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance

Noise points have the k^{th} nearest neighbor at farther distance

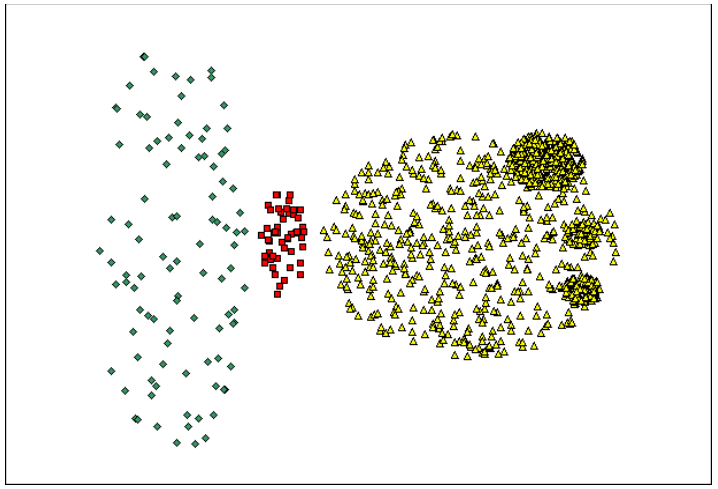
So, plot sorted distance of every point to its k^{th} nearest neighbor



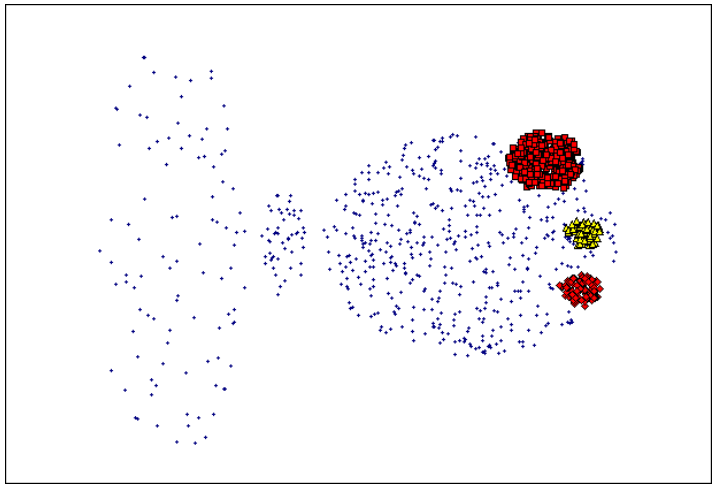
When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.75).



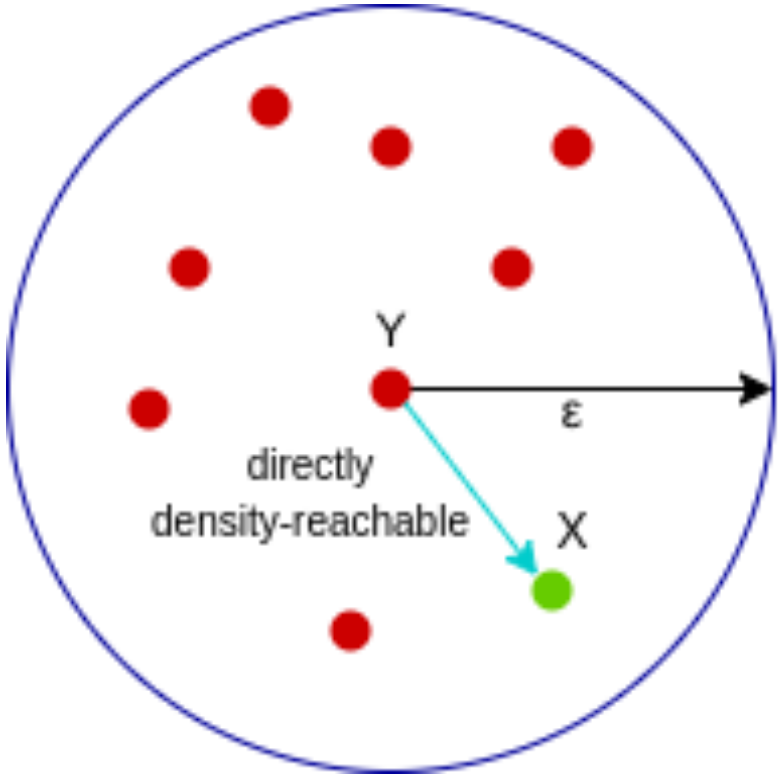
(MinPts=4, Eps=9.92)

- Varying densities
- High-dimensional data

Tan, Steinbach and Kumar, "Introduction to Data Mining"

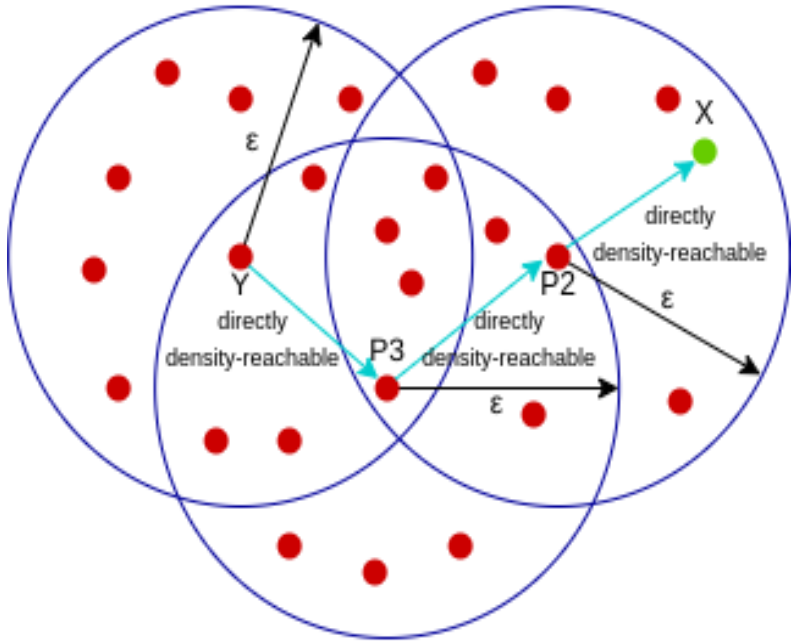
Reachability and Connectivity

- X belongs to the neighborhood of Y, i.e, dist(X, Y) <= epsilon
- Y is a core point
- Here, X is directly density-reachable from Y, but vice versa is not valid.



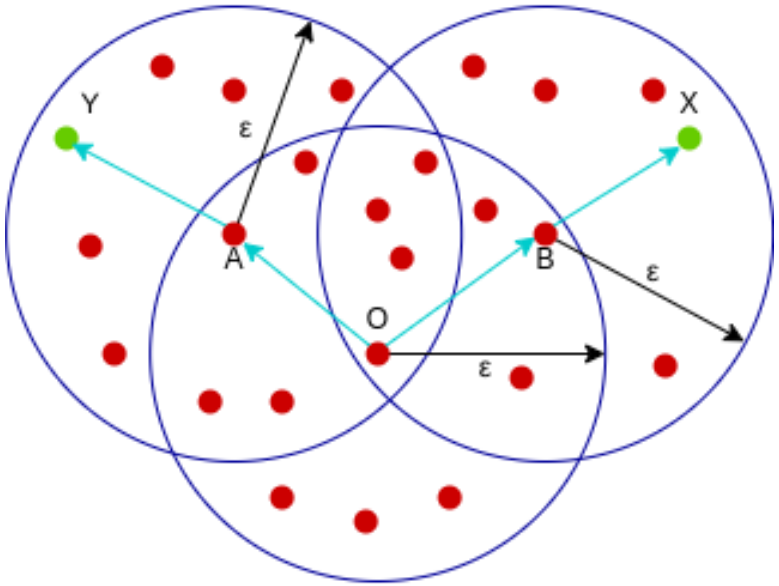
Reachability and Connectivity

- A point X is density-reachable from point Y w.r.t epsilon, minPoints,
- if there is a chain of points p1, p2, p3, …, pn and p1=X and pn=Y such that pi+1 is directly density-reachable from pi.
- Here, X is density-reachable from Y with X being directly density-reachable from P2, P2 from P3, and P3 from Y. But, the inverse of this is not valid.



Reachability and Connectivity

- A point X is density-connected from point Y w.r.t epsilon and minPoints.
- if there exists a point O such that both X and Y are density-reachable from O w.r.t to epsilon and minPoints.
- Here, both X and Y are density-reachable from O, therefore, we can say that X is density-connected from Y.



Parameter Selection in DBSCAN Clustering

- DBSCAN is very sensitive to the values of epsilon and minPoints.
- Therefore, it is very important to understand how to select the values of epsilon and minPoints.
- A slight variation in these values can significantly change the results produced by the DBSCAN algorithm.
- The value of minPoints should be at least one greater than the number of dimensions of the dataset, i.e.,
  - minPoints>=Dimensions+1.
- It does not make sense to take minPoints as 1 because it will result in each point being a separate cluster.
  - Therefore, it must be at least 3.

Parameter Selection in DBSCAN Clustering

- Generally, MinPts is set to twice the dimensions.

  But domain knowledge also decides its value.
- The value of epsilon can be decided from the K-distance graph.
- The point of maximum curvature (elbow) in this graph tells us about the value of epsilon.
- If the value of epsilon chosen is too small then a higher number of clusters will be created, and more data points will be taken as noise.
- Whereas, if chosen too big then various small clusters will merge into a big cluster, and we will lose details.

Parameter Selection in DBSCAN Clustering

- https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

Cluster validation

Cluster Cohesion or compactness: Measures how closely related are objects in a cluster

Example: SSE

Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters

Example: Squared Error

Cohesion is measured by the within cluster sum of squares (WSS)

$$WSS = \sum_{i} \sum_{x \in C_i} (x - m_i)^2$$

Separation is measured by the between cluster sum of squares (BSS)

$$BSS = \sum_{i} |C_i| (m - m_i)^2$$

Where |C_i| is the size of cluster i

Metrics for Measuring DBSCAN's Performance:

Silhouette Score: The silhouette score is calculated utilizing the mean intra- cluster distance between points, AND the mean nearest-cluster distance. For instance, a cluster with a lot of data points very close to each other (high density) AND is far away from the next nearest cluster (suggesting the cluster is very unique in comparison to the next closest), will have a strong silhouette score. A silhouette score ranges from -1 to 1, with -1 being the worst score possible and 1 being the best score. Silhouette scores of 0 suggest overlapping clusters.

References

Chapter 7.6 of R1 Han, Kamber and Pei (2nd Edn)
OR
Chapter 10.4 of R1 Han, Kamber and Pei (3rd Edn)

T1: Business Analytics, The Science of Data-Driven Making", U. Dinesh Kumar, Wiley 2017 (Chapter 14.4.4)

https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/

DATA ANALYTICS

Unit 4: Content Based Analysis- Dealing with Textual Data

Jyothi R.
Department of Computer Science and Engineering

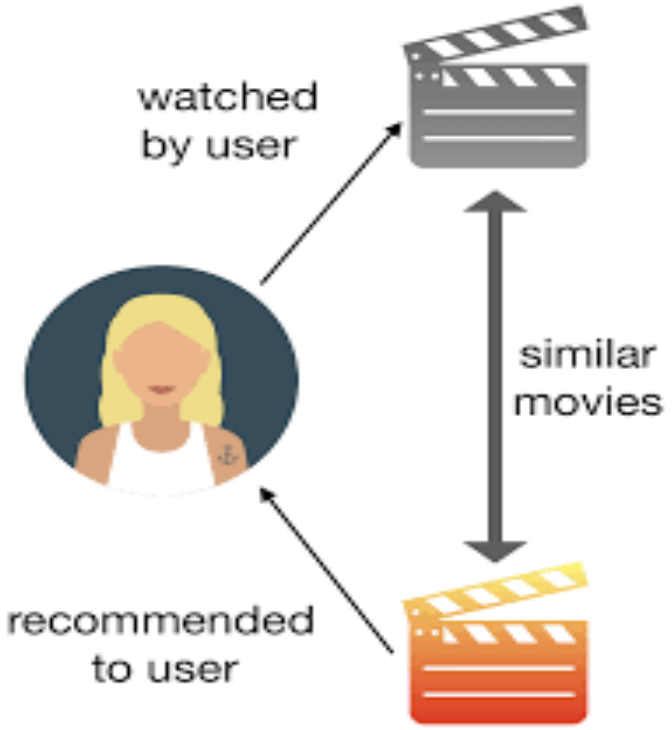
DATA ANALYTICS

Introduction

1. Basic Components of Content-Based Systems
2. Preprocessing and Feature Extraction
3. Examples

Introduction

- Content-based systems are designed to exploit scenarios in which items can be described with a descriptive sets of attributes (user profile, product features, etc.).
- And in such cases where a user's own ratings and actions on other movies are insufficient to discover meaningful recommendations.
- This approach is particularly useful when the item is new, and there are few ratings available for that item (cold start problem of collaborative filtering)



DATA ANALYTICS

Introduction

- Content-based recommender systems try to match users to items that are similar to what they have liked in the past.
- This similarity is not necessarily based on rating correlations across users but on the basis of the attributes of the objects liked by the user.
- Unlike collaborative systems, which explicitly leverage the ratings of other users in addition to that of the target user, content-based systems largely focus on the target user's own ratings and the **attributes of the items** liked by the user.
- Therefore, the other users have little, if any, role to play in content-based systems.
- In other words, the content-based methodology leverages a different source of data for the recommendation process.

DATA ANALYTICS

Content-Based Recommender Systems

- At the most basic level, content-based systems are dependent on two sources of data:

  1. The first source of data is a description of various items in terms of content-centric attributes.

  Example of such a representation could be the text description of an item by the manufacturer.

  2. The second source of data is a user profile, which is generated from user feedback about various items.

- The user feedback might be explicit or implicit. Explicit feedback may
- correspond to ratings, whereas implicit feedback may correspond to user actions.

DATA ANALYTICS

Content-Based Recommender Systems

- In a Content-based systems the ratings are collected in a way similar to collaborative systems.
- The user profile relates the attributes of the various items to user interests
- A very basic example of a user profile might simply be a set of labeled training documents of item descriptions, the user ratings as the labels, and a classification or regression model relating the item attributes to the user ratings.
- The specific user profile is heavily dependent on the methodology at hand. For example, explicit ratings might be used in one setting, and implicit feedback might be used in another.
- It is also possible for the user to specify her own profile in terms of keywords of interest,

Content-Based Recommender Systems

- Content-based systems are largely used in scenarios in which a significant amount of attribute information is available at hand.
- In many cases, these attributes are keywords, which are extracted from the product descriptions.
- In fact, the vast majority of content based systems extract text attributes from the underlying objects.
- Content-based systems are, therefore, particularly well suited to giving recommendations in text-rich and unstructured domains.
- A classical example of the use of such systems is in the recommendation of Web pages.
- For example, the previous browsing behavior of a user can be utilized to create a content-based recommender system.

DATA ANALYTICS

Content-Based Recommender Systems

- The use of such systems is not restricted only to the Web domain.
- Keywords from product descriptions are used to create item and user profiles for the purposes of recommendations in other e-commerce settings.
- In other settings, relational attributes such as manufacturer, genre, and price, may be used in addition to keywords.
- Such attributes can be used to create structured representations, which can be stored in a relational database.
- In these cases, it is necessary to combine the structured and unstructured attributes in a single structured representation.
- The basic principles of content-based systems, however, remain invariant to whether a structured or unstructured representation is used.

How do Content Based Recommender Systems work?

- A content based recommender works with data that the user provides, either explicitly (rating) or implicitly (clicking on a link).
- Based on that data, a user profile is generated, which is then used to make suggestions to the user. As the user provides more inputs or takes actions on the recommendations, the engine becomes more and more accurate.

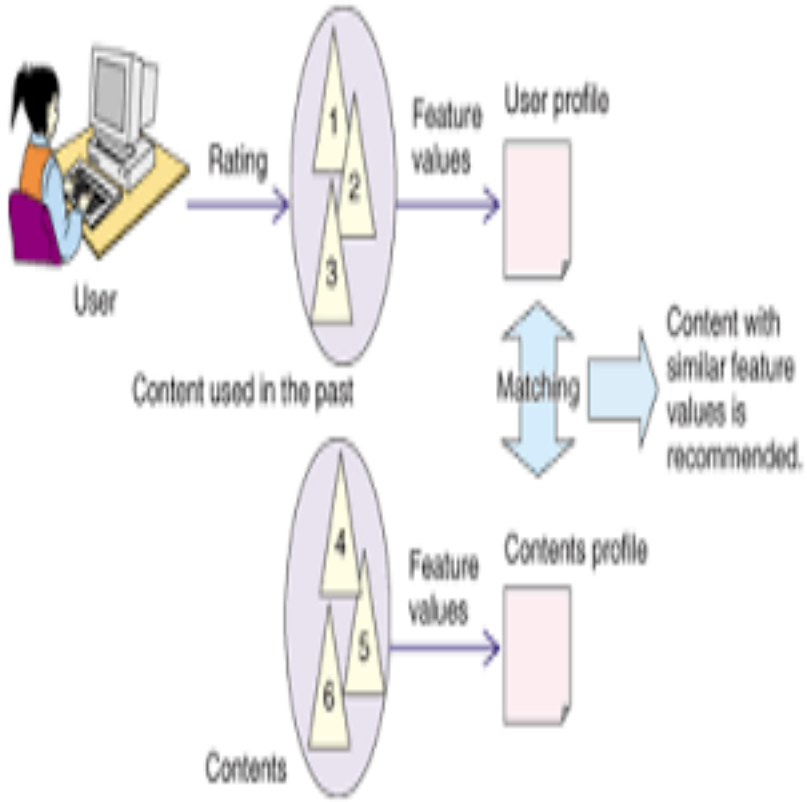
DATA ANALYTICS

How do Content Based Recommender Systems work?

1. Preprocessing and feature extraction: Features are extracted from various sources to convert them into a keyword-based vector-space representation. This is the first step of any content based recommendation system, and it is highly domain-specific. However, the proper extraction of the most informative features is essential for the effective functioning of any content-based recommender system.

2. Content-based learning of user profiles: A user-specific *model* is constructed to predict user interests in items, based on their past history of either buying or rating items. In order to achieve this goal, user feedback is leveraged, which may be manifested in the form of previously specified ratings (explicit feedback) or user activity (implicit feedback).. A learning model is constructed on this training data. This stage is often not very different from classification or regression modeling, depending on whether the feedback is categorical (e.g., binary act of selecting an item), or whether the feedback is numerical (e.g., ratings or buying frequency). The resulting model is referred to as the *user profile* because it conceptually relates user interests (ratings) to item attributes.

3. *Filtering and recommendation:* In this step, the learned model from the previous step is used to make recommendations on items for specific users. It is important for this step to be very efficient because the predictions need to be performed in real time.



http://findoutyourfavorite.blogspot.com/2012/04/content-based-filtering.html

Example: Movie Recommendation

Consider a movie recommendation site such as IMDb, that provides personalized recommendations for movies. Each movie is usually associated with a description of the movie such as its synopsis, the director, actors, genre, and so on. A short description of *Shrek* at the IMDb Website is as follows:

"After his swamp is filled with magical creatures, an ogre agrees to rescue a princess for a villainous lord in order to get his land back."

Many other attributes, such as user tags, are also available, which can be treated as content centric keywords. In the case of *Shrek*, one might simply concatenate all the keywords in the various fields to create a text description. The main problem is that the various keywords may not have equal importance in the recommendation process. For example, a particular actor might have greater importance in the recommendation than a word from the synopsis. This can be achieved in two ways:

1. Domain-specific knowledge can be used to decide the relative importance of keywords.
For example, the title of the movie and the primary actor may be given more weight than the words in the description. In many cases, this process is done in a heuristic way with trial and error.

2. In many cases, it may be possible to *learn* the relative importance of various features in an automated way. This process is referred to as feature weighting, which is closely related to feature selection. Both feature weighting and feature selection are described in a later section.

DATA ANALYTICS

How do we convert text to numbers for content based recommendation?

- **What are the concepts used in Content Based Recommenders?**

- The concepts of Term Frequency (TF) and Inverse Document Frequency (IDF) are used in information retrieval systems and also content based filtering mechanisms (such as a content based recommender).
- They are used to determine the relative importance of a document / article / news item / movie etc.

DATA ANALYTICS

How do we convert text to numbers for content based recommendation?

Term Frequency (TF) and Inverse Document Frequency (IDF)

- TF is simply the frequency of a word in a document.

$$\text{Tf}(t) = \frac{\text{Frequency occurence of term t in document}}{\text{Total number of terms in document}}$$

- IDF is the inverse of the document frequency among the whole corpus of documents.

$$\text{Idf}(t) = log_{10}(\frac{\text{Total Number of documents}}{\text{Number of documents containing term t}})$$

- TF-IDF is used mainly because of two reasons: Suppose we search for “the rise of analytics” on Google.
- It is certain that “the” will occur more frequently than “analytics” but the relative importance of analytics is higher than the search query point of view.

  In such cases, TF-IDF weighting negates the effect of high frequency words in determining the importance of an item (document).

DATA ANALYTICS

How do we convert text to numbers for content based recommendation?

Term Frequency (TF) and Inverse Document Frequency (IDF)

Example:

Consider a document containing 100 words wherein the word *cat* appears 3 times. The term frequency (i.e., tf) for *cat* is then (3 / 100) = 0.03. Now, assume we have 10 million documents and the word *cat* appears in one thousand of these. Then, the inverse document frequency (i.e., idf) is calculated as log(10,000,000 / 1,000) = 4. Thus, the Tf-idf weight is the product of these quantities: 0.03 * 4 = 0.12.

DATA ANALYTICS

How do we convert text to numbers for content based recommendation?

Term Frequency (TF) and Inverse Document Frequency (IDF)

- But while calculating TF-IDF, log is used to dampen the effect of high frequency words.
- For example: TF = 3 vs TF = 4 is vastly different from TF = 10 vs TF = 1000.
- In other words the relevance of a word in a document cannot be measured as a simple raw count and hence may incorporate weights as shown below:

$$w_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d}, & \text{if } tf_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

Term Frequency	Weighted Term Frequency
0	0
10	2
1000	4

Inverse document frequency (IDF)

$\log_{10}(n/n_i)$ where n is the total number of documents and

n_i the number of documents in which the term i^{th} term appears

TF-IDF is a product of the term frequency and inverse document frequency

DATA ANALYTICS

How do we convert text to numbers for content based recommendation?

- It can be seen that the effect of high frequency words is dampened and these values are more comparable to each other as opposed to the original raw term frequency.
- Hence the computation of TF-IDF is preceded by
  - Stop word removal (eliminating articles (a, an, the), prepositions, conjunctions, and pronouns)
  - Stemming (hoping, hoped, hope -> the root word 'hop')

    (this can be detrimental as hope and hop (jump) can be confused with each other)
  - Phrase extraction ("cross examine" can mean something different from 'cross' or 'examine')
- After calculating TF-IDF scores, how do we determine which items are closer to each other, rather closer to the user profile?
- This is accomplished using the Vector Space Model which computes the proximity based on the angle between the vectors.

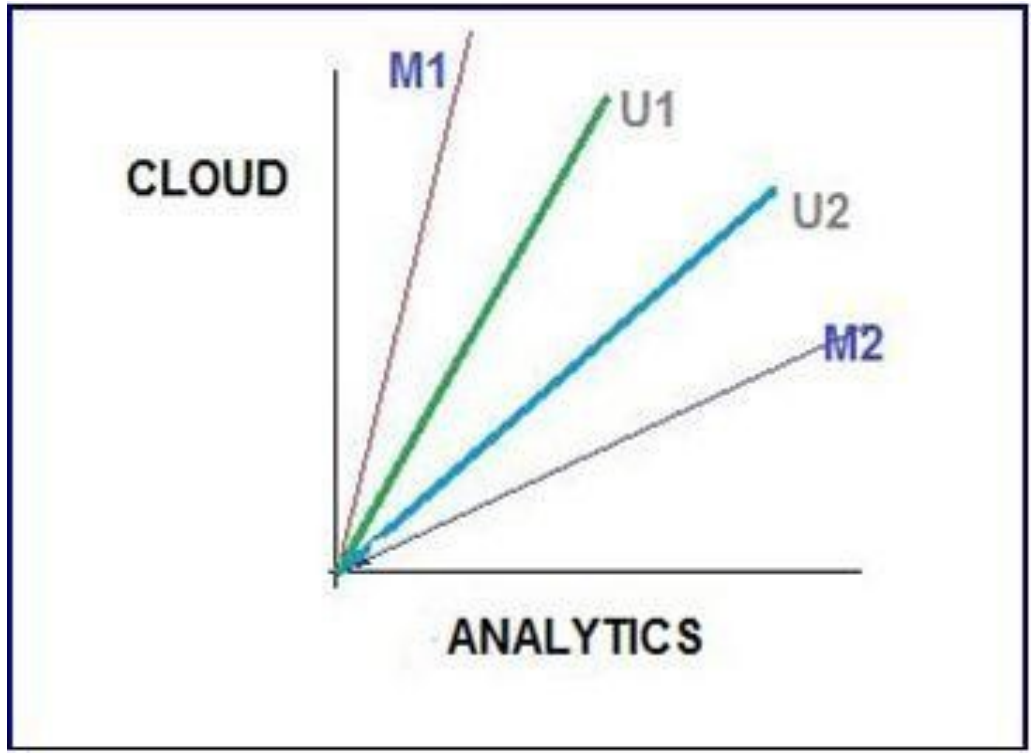
How does a Vector Space Model work?

- In this model, each item is stored as a vector of its attributes (which are also vectors) in an n-dimensional space and the angles between the vectors are calculated to determine the similarity between the vectors.
- Next, the user profile vectors are also created based on his actions on previous attributes of items and the similarity between an item and a user is also determined in a similar way.

DATA ANALYTICS

How does a Vector Space Model work?

Lets try to understand this with an example.



- Shown above is a 2-D representation of a two attributes, Cloud and Analytics.
- M1 & M2 are documents.
- U1 & U2 are users.
- The document M2 is more about Analytics than cloud whereas M1 is more about cloud than Analytics.
- User U1, likes articles on the topic 'cloud' more than the ones on 'analytics' and vice-versa for user U2.
- The method of calculating the user's likes / dislikes / measures is calculated by taking the cosine of the angle between the user profile vector(Ui) and the document vector.

DATA ANALYTICS

Case study – How do we calculate TF – IDF ?

- Let's understand this with an example. Suppose, we search for "IoT and analytics" on Google and the top 5 links that appear have some frequency count of certain words as shown below:

Articles	Analytics	Data	Cloud	Smart	Insight
Article 1	21	24	***0***	2	2
Article 2	24	59	2	1	***0***
Article 3	40	115	8	10	19
Article 4	4	28	5	***0***	1
Article 5	8	48	4	3	4
Article 6	17	49	8	***0***	5
DF	5,000	50,000	10,000	5,00,000	7000

- Among the corpus of documents / blogs which are used to search for the articles, 5000 contains the word analytics, 50,000 contain data and similar count goes for other words. Let us assume that the total corpus of docs is 1 million(10^6).

DATA ANALYTICS

Term Frequency (TF)

- As seen in the image below, for article 1, the term Analytics has a TF of 1+ log1021 = 2.322. In this way, TF is calculated for other attributes of each of the articles. These values make up the attribute vector for each of the articles.

Articles	Analytics	Data	Cloud	Smart	Insight	Length of Vector
Article 1	2.322219295	2.380211242	0	1.301029996	1.301029996	**3.800456039**
Article 2	2.380211242	2.770852012	1.301029996	1	0	**4.004460697**
Article 3	2.602059991	3.06069784	1.903089987	2	2.278753601	**5.380804488**
Article 4	1.602059991	2.447158031	1.698970004	0	1	**3.527276247**
Article 5	1.903089987	2.681241237	1.602059991	1.477121255	1.602059991	**4.257450611**
Article 6	2.230448921	2.69019608	1.903089987	0	1.698970004	**4.326697114**

How does Vector Space Model works?

- The reason behind using cosine is that the value of cosine will increase with decreasing value of the angle between which signifies more similarity.
- The vectors are length normalized after which they become vectors of length 1 and then the cosine calculation is simply the sum-product of vectors.

References

Text Book:

"Business Analytics, The Science of Data-Driven Making", U. Dinesh Kumar, Wiley 2017

"Recommender Systems, The text book, Charu C. Aggarwal, Springer 2016 Chapter 4

Image Courtesy

https://towardsdatascience.com/introduction-to-two-approaches-of-content-based-recommendation-system-fc797460c18c

https://www.kaggle.com/ashish95arora/content-based-recommender-using-text-mining

THANK YOU

Jyothi R.
Assistant Professor,
Department of Computer Science
jyothir@pes.edu