Effective implementation of hierarchical clustering

IOP Conf. Series: Materials Science and Engineering **263** (2017) 042094 doi:10.1088/1757-899X/263/4/042094

Effective implementation of hierarchical clustering

Mudita Verma, V Vijayarajan, G Sivashanmugam and D Geraldine Bessie Amali
School of Computer Science and Engineering, VIT University, Vellore-632014, India

E-mail: vijayarajan.v@vit.ac.in

Abstract. Hierarchical clustering is generally used for cluster analysis in which we build up a hierarchy of clusters. In order to find that which cluster should be split a large amount of observations are being carried out. Here the data set of US based personalities has been considered for clustering. After implementation of hierarchical clustering on the data set we group it in three different clusters one is of politician, sports person and musicians. Training set is the main parameter which decides the category which has to be assigned to the observations that are being collected. The category of these observations must be known. Recognition comes from the formulation of classification. Supervised learning has the main instance in the form of classification. While on the other hand Clustering is an instance of unsupervised procedure. Clustering consists of grouping of data that have similar properties which are either their own or are inherited from some other sources.

1. Introduction

Learning algorithms are of two types supervised and unsupervised learning. In supervised learning, we provide labels and our documents or items in the cluster go to a particular label. Thus, if we have news report as the dataset I can classify it as sports politics or business based on the label in the case of supervised learning due to the presence of specific labels. However, in unsupervised learning there are no labels. We can create clusters, where similar articles can be clustered together. There are however no labels, and no limit on the number of clusters in the clustering algorithm. In unsupervised clustering algorithm, we can use the distance between two sets of articles to a particular cluster to determine which document is nearer to which cluster. Classification widely used in recommender systems, especially in online shopping where products which have one name or one type of genre, can be clustered and shown together to show to the user. Clustering is widely used in various e commerce companies like amazon.com or online music or movie streaming companies like Netflix.com or Spotify, which based on user's choices cluster the related movies or songs and recommend to the interested user. Clustering systems are also used for images or computational photography where images belonging to a particular type like photos of books, Similar faces can be clustered together. And shown to user, another important application of clustering is that advertisements can only be show to specific set of users instead of showing to an entire lot of people.

2. Literature Survey

Researchers optimize Supervised and Unsupervised learning have approaches as to the implementation of classification system, the ones most relevant to modern approach of building a classification system, the ones most approach of building a classification system, the ones classification algorithm and Structural Equation using publication work and organizing its text snippets under the lables such as: title, abstract, semi structure, medata. This activity aids in Text classification across pdf files. [1] Moreover Semi supervised learning using multiple Clustering with

limited labelled data is used as normally any machine learning based algorithm produces good results or make good predictive models when more data is its training data. [3] However in case of supervised we need labelled data to train our model on, but due to lack of labelled training data (supervised learning) and abundance of unlabelled data (unsupervised learning),models are developed which could learn from both labelled and unlabelled data. [4] As learning is part of creating a system that works for Humans and their behaviour**.** Semantic analysis is used by the classification of the social, short text which are taken from the entropy model with maximum values which are obtained from very short sentences such as tweets, news headlines, comments, reviews, to improve an organization services Topic level entropy or TIME focusses on social Emotion classification over short texts.[7] There various resources that are present that can consist of the following:

A) Class based semantics for hybrid semi supervised algorithm for text Classification with class based semantics.
B) Data stream modification with the help of incremental machine learning algorithm.
C) Machine learning for data that is spatial.
D) Equation modelling for structure depends upon the satisfaction of customers as well as the analysis of the customer loyalty.

Machine Learning can be used to divide and classify any given set of data under a label and can be Extremely precise about it to increase efficiency of useful data retrieval [8].

3. Algorithm

3.1 Word Count:

The articles are listed and a word count vector is created. The word count vector is basically an array which keeps the count of number of words appeared in an article. data (unsupervised learning), models are developed. A simple word count vector may look like below Figure 1 and Figure 2.

word	*count*
The	40
in	30
and	21
of	18

Figure 1. The above words are most frequent occuring words

word	*count*
normalize	1
sought	1
combat	1
unconstitutional	1

Figure 2. The above given words are the least occuring words

3.1.1 Demerits of Word Count:

The disadvantage of the word count vector clustering is that words such as “and”,”the”,”a” “an” are given quite a high weightage which quite inconclusive to label or cluster the data but important words

(which do not occur frequently) are not given due importance. So, clustering using this algorithm is produces trival and inconclusive results, even though this algorithm is extremely simple build.

3.2. K-Nearest Neighbour:

K-Nearest Neighbour(KNN) is a simple machine learning approach. KNN is easily approached because of its simplicity and flexibility to process different data variables. The objectivity of KNN is to estimate on a fixed no. of observation, say k, which is nearest to the desired outcome. Here we use nearest neighbour search to determine which articles are nearer to each other. This is useful especially in unsupervised learning where we do have labels and we only need to cluster similar documents. It doesn't build any model or function, but still, it predicts the nearest k records from the training data set that is including the most similarity to the test. KNN is referred as lazy learning algorithm because of this nature. It can be implemented on both continuous and discrete known as regression and classification respectively. Regression carries out the k neighbour average, whereas classification carries out a frequent neighbour. It is a supervised algorithm, training data n pair (xi , yi) and y(x) is to find out the problem from a new input x. To implement the technique, it is required to have a training set and a test sample. In order to know the value of k, and the formula for the distance between the instances**.** The nearest neighbour with similar characteristics is given by:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 = \arg\min_{S} \sum_{i=1}^{k} |S_i| \, Var \; S_i$$

3.2.1. Algorithm:

for all test example x do

for all training example (x_i, y_i) *, do*

compute distance (x, x_i) *;*

end for

select the k-nearest neighbour of x;

return the average output value among neighbours i.e. $1/k \sum_i^k y_i$ *;*

end for;

3.3 TF-IDF:

TF-IDF i.e. "Term Frequency, Inverse Document Frequency" is an approach to score the significance of words (or "terms") in a record in light of how as often as possible they show up over different archives. On the off chance that a word shows up as often as possible in a report, it's vital. Give the word a high score. In any case, if a word shows up in many records, it's not an interesting identifier. Give the word a low score. Subsequently, regular words like "the" and "for", which show up in many archives, will be downsized. Words that show up as often as possible in a solitary report will be scaled up.

TF-IDF is the result of term recurrence and converse record recurrence.

3.3.1. Term Frequency:

Term frequency tf(t,d) represents how many time t happens to occur in d. This is given by :

$$tf(t,d) = 0.5 + 0.5 \cdot \frac{f_{t,d}}{\max\{f_{t',d} : t' \in d\}}$$

3.3.2. Inverse Document Frequency:

The converse archive recurrence speaks to the measure of data given by that particular word. It additionally tells whether the term is normal or uncommon in the accessible archives. Converse report recurrence is spoken to by idf(t,D) and N gives the aggregate number of archives. Idf can be given by:

$$idf(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

3.3.3. Term Frequency - Inverse Document Frequency:
Then tf-idf can be calculated as:

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D)$$

4. Mathematical Example
Consider two counting tables consisting only two documents as shown below Table 1 and Table 2.

Table 1. Training Accuracy of NN Models in percentages:

Text File 1	
Term	Term Count
That	2
This	1
Is	2
A	1

Table 2. Training Accuracy of NN Models in percentages:

Text File 2	
Term	Term Count
That	2
A	3
This	1
Consider	1

Here the tf-idf for the term “this” is being calculated.

Term frequency is given by:

$$tf(\text{"this"}, d_1) = \frac{1}{5} = 0.2$$

$$tf(\text{"this"}, d_2) = \frac{1}{7} \approx 0.14$$

Inverse document frequency is given by:

$$idf(\text{"this"}, D) = \log\left(\frac{2}{2}\right) = 0$$

In the event that tf-idf has esteem zero for a specific word then this infers the word is very little educational as it shows up in the given records.

$$tfidf(\text{"this"}, d_1) = 0.2 \text{ x } 0 = 0$$

$$tfidf(\text{"this"}, d_2) = 0.14 \text{ x } 0 = 0$$

5. Dataset

Dataset used here is the list of famous US Based personalities and the information that Wikipediae has about them. Here hierarchical clustering is being applied on these personalities in such a way that a particular cluster for the politicians, musicians and sports persons is being created. A total of 59071 personalities is being considered which would be clustered, and the similar personalities will be brought together.

6. Result

Here the similarity would be formed with word count method as well as TF-IDF method. Further on the basis of the results obtained, the preferred model would be suggested.

6.1 Word Count

Consider the example of Victoria Beckham. According to the word count model, the most similar personality to Victoria beckham is Mary fitzgerald. But we know that this clustering is not that relevant to the clusters that are to be formed. Hence, we say that the word count analysis is not appropriate. Here the frequency is divided by the total number of tokens, i.e. word occurrences, in the training set. The reason is if we divide by the number of distinct words, the probabilities for all words will not necessarily sum to one so they won't form a probability distribution. The word count result is shown in Figure 3.

```
word_count_model.query(victoria)
```

```
Starting pairwise querying.
+--------------+---------+-------------+--------------+
| Query points | # Pairs | % Complete. | Elapsed Time |
+--------------+---------+-------------+--------------+
| 0            | 1       | 0.00169288  | 0us          |
| Done         |         | 100         | 390.631ms    |
+--------------+---------+-------------+--------------+
```

query_label	reference_label	distance	rank
0	Victoria Beckham	-2.22044604925e-16	1
0	Mary Fitzgerald (artist)	0.207307036115	2
0	Adrienne Corri	0.214509782788	3
0	Beverly Jane Fry	0.217466468741	4
0	Raman Mundair	0.217695474992	5

[5 rows x 4 columns]

Figure 3 Word Count Result

6.2 TF-IDF

According to TF-IDF method, the result shows that the most relevant or similar personality to Victoria backham is her husband david beckham, then their children, and so on. Hence TF-IDF has taken into account the most frequent as well as the most relevant character that would help in keeping the personality in a specific cluster. Not all the models work in such a manner. Hence, we are able to cluster in clusters as decided by us. The TF-IDF result is shown in Figure 4.

```
tfidf_count_model.query(victoria)
```

```
Starting pairwise querying.
+--------------+---------+-------------+--------------+
| Query points | # Pairs | % Complete. | Elapsed Time |
+--------------+---------+-------------+--------------+
| 0            | 1       | 0.00169288  | 31.25ms      |
| Done         |         | 100         | 453.13ms     |
+--------------+---------+-------------+--------------+
```

query_label	reference_label	distance	rank
0	Victoria Beckham	1.11022302463e-16	1
0	David Beckham	0.548169610263	2
0	Stephen Dow Beckham	0.784986706828	3
0	Mel B	0.809585523409	4
0	Caroline Rush	0.819826422919	5

[5 rows x 4 columns]

Figure 4. TF-IDF Result

7. Conclusion

Comparison of the above two algorithms on the basis of similar characteristics of the personalities, shows that TF-IDF model is best suitable of hierarchical clustering. Tf-idf provides the clusters which has the people who are genuinely connected to each other. But on the other hand, word count model give us the cluster on the basis of unimportant selection of words. It picks up the most frequent words irrespective of their relevance with the personality. On the other hand, TF-IDF selects those words which have proper relevance as well as have more frequency in the given text.

References

[1] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville 2016 Deep learning MIT Press
[2] Tang, Yichuan 2013 Deep learning using linear support vector machines arXiv preprint arXiv: 1306.0239
[3] Li, Jiwei 2015 Visualizing and understanding neural models in nlp" arXiv preprint arXiv: 1506. 01066
[4] Bengio, Yoshua 2009 Learning deep architectures for AI Foundations and trends® in Machine Learning 2.1 1-127
[5] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton 2015 Deep learning Nature 521.7553 436-444
[6] Glorot, Xavier, Antoine Bordes, and Yoshua Bengio 2011 Deep Sparse Rectifier Neural Networks Aistats. Vol. 15. No. 106
[7] LeCun, Yann 2015 LeNet-5, convolutional neural networks URL: http://yann. lecun. com/exdb/lenet
[8] Schmidhuber, Jürgen 2015 Deep learning in neural networks: An overview Neural networks 61 85-117