*****

RAMAKRISHANA MISSION VIVEKANANDA EDUCATIONAL AND RESEARCH INSTITUTE

BELUR MATH, HOWRAH, 711202 WB

Institute Name :

SCHOOL OF MATHEMATICAL SCIENCE

DEPARTMEN NAME : COMPUTER SCIENCE

Assignment : 2

Submited By/Name : JEETU KUMAR
Registration No./Roll No. : B18732
Program Name : M.Sc. Big Data Analytics
Semester : 2nd, January - May
Paper : Machine Learning
Instructor/Paper-Instructor : Dr. Tanmay Basu

April 28, 2019

Assignment Release Date: April 04, 2019
Due Date Of Submission: April 28, 2019

1 WORD CLUSTERING

1 Introduction

Lets introduce the data set. Data set given having 38 comments as a answer of question ***"What qualities do you think are necessary to be the prime minister of India?",*** comments suppose to having answer or specification on stated question. These comments having *english words and texts in english alphabets.* Our task is to find 'cluster(grouping)'[1] in comments using 'WordNet'[11], 'NLTK'[10], and appropriate clustering algorithm for 'Scikit-Learn'[7] library, and required pre-processing.
WordNet is a large lexical database for the English language. It groups English words into sets of synonyms called *synsets*, provides short definitions and usage examples, and records a number of relations among these synonym sets or their members. WordNet can thus be seen as a combination of dictionary and thesaurus.
WordNet includes the lexical categories nouns, verbs, adjectives and adverbs but ignores prepositions, determiners and other function words.

2 Methodology

2.1 Key Terms Form Language

A **Synonym** is a word or phrase that means exactly or nearly the same as another lexeme in the same language. Words that are synonyms are said to be synonymous, and the state of being a synonym is called synonymy. For example, the words begin, start, commence, and initiate are all synonyms of one another Also, sometimes the word you have in mind might not be the most appropriate word, which is why finding the right synonym can come in handy.
Antonym a word opposite in meaning to another (e.g. bad and good).

Hyponym[12][11] a word of more specific meaning than a general or superordinate term applicable to it. For example, spoon is a hyponym of cutlery.
hyponyms: Y is a hyponym of X if every Y is a (kind of) X (dog is a hyponym of canine)[12]
Hypernym [11][12] a word with a broad meaning constituting a category into which words with more specific meanings fall; a superordinate. For example, colour is a hypernym of red.
hypernyms: Y is a hypernym of X if every X is a (kind of) Y (canine is a hypernym of dog)[12]
In linguistics, a hyponym is a word or phrase whose semantic field is included within that of another word, its hyperonym or hypernym. In simpler terms, a hyponym is in a type-of relationship with its hypernym.

Lemma (Morphology)[14][11] a heading indicating the subject or argument of a literary composition or annotation.
A lemma is the word you find in the dictionary. A lexeme is a unit of meaning, and can be more than one word. A lexeme is the set of all forms that have the same meaning, while lemma refers to the particular form that is chosen by convention to represent the lexeme.
Morphological derivation (Derivational Form)[13] , in linguistics, is the process of forming a new word from an existing word, often by adding a prefix or suffix, such as -ness or un-. For example, happiness and unhappy derive from the root word happy.
It is differentiated from inflection, which is the modification of a word to form different grammatical categories without changing its core meaning: determines, determining, and determined are from the root determine.
Derivation can be contrasted with inflection, in that derivation produces a new word (a distinct lexeme), whereas inflection produces grammatical variants of the same word.

Words from the same lexical category that are roughly synonymous are grouped into **Synset[11]** . Synsets include simplex words as well as collocations like "eat out" and "car pool." The different senses of a polysemous word form are assigned to different synsets. The meaning of a synset is further clarified with a short defining gloss and one or more usage examples. An example adjective synset is:
good, right, ripe – (most suitable or right for a particular purpose; "a good time to plant tomatoes"; "the right time to act"; "the time is ripe for great sociological changes")
All synsets are connected to other synsets by means of semantic relations. These relations, which are not all shared by all lexical categories, include:
Nouns(hypernyms, hyponyms, coordinate terms, meronym, holonym)[11][12]
Verbs(hypernym, troponym, entailment, coordinate terms)[11][12]

2.2 K-Mean

The commonly used partitional clustering technique is k-means method, where k is the desired number of clusters [1]. The algorithm initially chooses k number of points randomly from the data set,which are called as seed points. Each point is assigned to its nearest seed point,thereby creating k clusters. Therefore the centroids of the clusters are computed and each data point is assigned to its nearest centroid. In general,the centroid of a cluster is considered as the mean of the data points of the cluster. Hence the method is known as k-means algorithm. Subsequently,the algorithm groups all the data points to their nearest centroids and again computes the cluster centroids and so on. The method stops when the centroids of the clusters for two consecutive iterations runs are same.

2.3 Hierachical Clustering

The hierarchical clustering techniques produce a hierarchical tree of clusters known as dendogram [1]. The hierarchical clustering algorithms need not require the number of clusters prior implementation. There are two types of hierarchical

clustering techniques - agglomerative and divisive [1]. The basic steps of the agglomerative clustering algorithms are as follows:

- Consider each feature vector in the dataset as a singleton cluster.
- Find the distance between each pair of clusters.
- Find the pair of minimal distant clusters and merge them to form a new cluster. Therefore the number of clusters is reduced by one.
- Repeat step 2 and step 3 until all feature vectors are grouped in a single cluster. The method may also be terminated if the desired number of clusters are obtained at any level of the dendogram. In that case, we need to know the desired number of clusters prior implementation.

The second step is the most important part of any agglomerative hierarchical clustering technique. There are mainly three variations of agglomerative techniques based on how to find the distance between a pair of clusters. The methods are single linkage, complete linkage and group average agglomerative clustering techniques.

2.4 Directly Putting Threshold On Similarity Value

Here we will use *wup_similariry* and *path_similariry* from *wordnet* to get similarity score between two words of vocabulary(between two clusters). We find similarity score between *synsets* of two cluster(synsets of cluster means set of *synset* of 'words (including their *derivational* forms)' in a cluster), if these similarity score between two cluster satisfy some criteria then we will merge them else not and we will look for this merge until there is no new merge take place. i.e. if in i^{th} iteration number cluster is same as in $(i-1)^{th}$ and all merge possibility is over then we will stop.

2.5 Pre-Processing

- **D1** We will start with creating list of words by splitting the comments into the words, stop word removal and POS tagging.
  I have created three different list of words by considering POS preference as ('Noun, Adjective, Adverb, Verb'), ('Noun, Adjective, Adverb') and ('Noun, Adjective') namely L, $L1$, and $L2$ of size 192, 175, and 168 respectively
- **D2** We will create vocabulary of unique words from list of words .Let V be the vocabulary set of size(cardinality) n where th element v_i of V is a word(more specifically word of our purpose after $D1$).

  Here I have preferred to create three different vocabulary by taking ('Noun, Adjective, Adverb, Verb'), ('Noun, Adjective, Adverb') and ('Noun, Adjective') namely V, $V1$, and $V2$ of size 132, 118, and 112 respectively.
- **D3** We will create similarity matrix for all variants of vocabulary to work with methodology described in section 2.2 and 2.3 .

Let the set S_l and S_k is the *synsets* of l^{th} and k^{th} element of V of size p and q respectively.Then we will get total pq similarity value for l^{th} and k^{th} word. Let SV be the list(or tuple) of size pq where r^{th} element of SV is similarity value between some $p^{'}$ and $q^{'}$ element of S_l and S_k respectively returned by similarity metric(namely *wup_similarity* or *path_similarity* in this work). Let *cnt* be count of greater than 0 numeric numbers (size except 0 or 'None') in SV.

Let SM be the our similarity matrix then $SM = (sm)_{i,j}$; $for\ all\ i = 1, 2, \ldots, n\ j = 1, 2, \ldots, n$; where we will get $(sm)_{i,j}$ as follows:

$$(sm)_{i,j} = (sm)_{j,i} = \begin{cases} max\,(SV) & if\ 0 < pq \leq 1.4\cdot\text{cnt} \\ \frac{sum(SV)}{cnt} & if\ 1.4\cdot\text{cnt} \leq pq \leq 2\cdot\text{cnt} \\ \frac{sum(SV)}{pq} & if\ 2\cdot\text{cnt} \leq pq \leq 2.7\cdot\text{cnt} \\ min\,(SV) & if\ 2.7\cdot\text{cnt} \leq pq \\ 0 & o.w. \end{cases} \quad (1)$$

- **D4** Now we will use these similarity matrix to form cluster by using methodology described in section 2.2 and 2.3 .
  For these method we will decide the number of cluster by looking on '*Elbow*' curve and '*Variance acceleration and decleration*' curve.

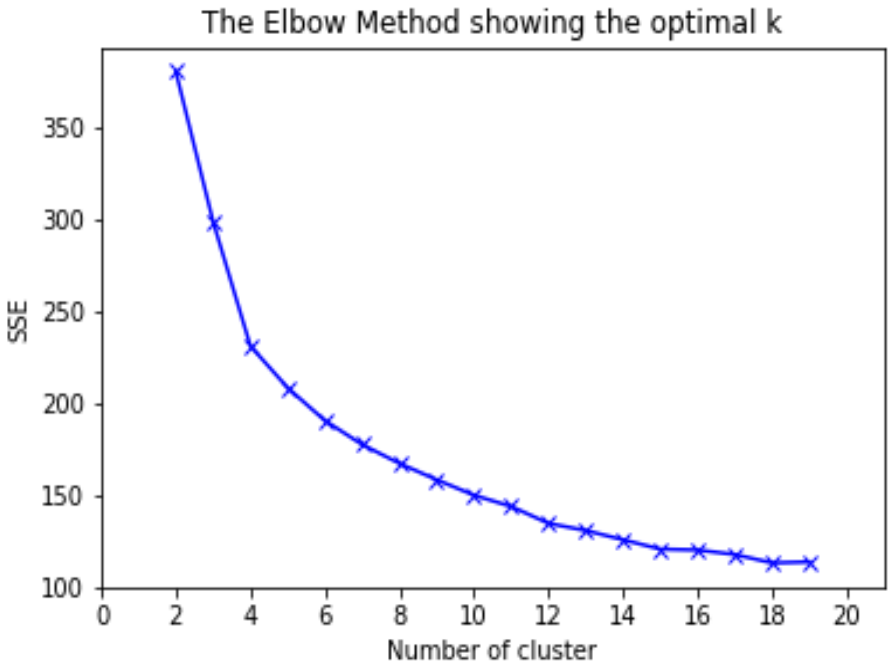

Figure 1: Elbow Of for KMean

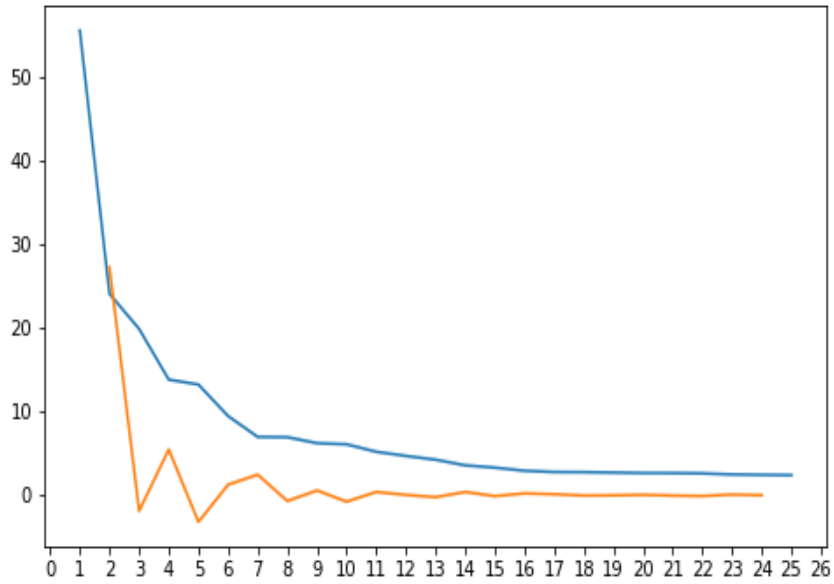

Figure 2: Elbow Of for linkage ward

By looking on '*Elbow*' curve and '*Variance acceleration anddecleration*' curve we can see that number cluster in human coding is too correct.
I have preferred 10 and 13 two number as number of cluster.

- **D5** In $D3$ and $D4$ we have discussed the way(map/pre-processing) for methodology described in section 2.2 and 2.3; For method described in section 2.4 we will follow the following steps sequentially.

  1. Start with making all word of vocabulary V as a singleton cluster. Let $C = \{C_1, C_2, \ldots, C_n\}$ where C_i has element v_i of V.
  2. The threshold a and b. Let $a = a^{'}$ and $b = b^{'}$
  3. Set $i = 0$, $j = i + 1$, and $n = n$
  4. Compute similarity values list SV for i^{th} and j^{th} cluster as follows: Let the set S_l and S_k is the *synsets* by all member of i^{th} and i^{th} element cluster of size p and q respectively.Then we will get total pq similarity value for i^{th} and j^{th} cluster, get these values to list(or tuple) SV of size pq where r^{th} element of SV is similarity value between some $p^{'}$ and $q^{'}$ element of S_l and S_k respectively returned by similarity metric(namely *wup_similarity* or *path_similarity*). Let *cnt* be count of greater than 0 numeric numbers (size except 0 or 'None') in SV .

5. Compute $ss_{i,j}$ and $av_{i,j}$ as follows:

$$(ss)_{i,j} = \begin{cases} max(SV) & if\ cnt \neq 0 \\ 0 & o.w. \end{cases} \tag{2}$$

$$(av)_{i,j} = \begin{cases} \frac{sum(SV)}{pq} & if\ cnt = pq \\ \frac{sum(SV)}{cnt} & if\ 0 < pq \leq 1.4 \cdot \text{cnt} \\ \frac{sum(SV)}{pq} & if\ 1.4 \cdot \text{cnt} \leq pq \leq 2 \cdot \text{cnt} \\ \frac{sum(SV)}{pq+cnt} & if\ 2 \cdot \text{cnt} \leq pq \\ 0 & o.w. \end{cases} \tag{3}$$

6. Repeat step 3 to 5 fro $j = i+1, \ldots, n$. Get $ss_{i,j}$ and $av_{i,j}$, if $av_{i,j} \geq a^{'}$ and $ss_{i,j} \geq b^{'}$ then add $ss_{i,j}$ to similarity score list SS.

7. $Part1$: If $SS \neq \Phi$ (empty set) set). Get maximum of SS. Let $max(SS) = ss_{i,j'}$ merge i^{th} and $j^{'\ th}$ cluster. Do update $i = i$, $j = i+1$ and $n = n-1$; Go to step 4.
   $Part2$: If $SS = \Phi$. Check $i == n$? if YES then $STOP$, if NO then do update $i = i+1$, $j = i+1$ and $n = n$; Go to step 4.

8. Step 7 tells $STOP$? then stop and return the current cluster.

3 Discussion And Conclusion

3.1 Analysis/Discussion Of Results

For Analysis/Discussion Of Results and to show all results of all experimental setup for all three vocabulary set and for different values of parameter $a^{'}$ and $b^{'}$ which I have described in section 2.5 , I will prefer to show CSV output file and $Data - Frame$ on jupyter notebook.

3.2 Results And Conclusion

Hierachical clustering works some what better then K-mean in transformed domain, for k-mean if we can provide number of cluster with notation of centriod(or only information of centriod of desired number of cluster) then for some specific it may work with desired accuracy in transformed domain; in else situation specific to our case there average performance of both does not differs more.
Number Of clusters specified in "Human coding" of sample output(provided by instructor) looks fully reasonable as through several run of k-mean and linkage-ward by plotting the '*Elbow*' curve and '*Variance acceleration and decleration*' curve, I have not got any more different result.

In case of direct threshold-ing of similarity-score proper-tuning of $a^{'}$ and $b^{'}$ (as defined in section 2.5) which is a hard task although range is (0 1) possibility is infinite. So it is lazy and time consuming. But it gives reasonable results in context of Human-Coding and result of other method.

From here reducing issues affecting the performance of k-mean and hierachical techniques in tarnsformed domain by getting proper distance function and and providing maximum possible evidence in pre-processing could be a good feature work.

Thanks!

The data have been shared by Instructor :
https : *//mail.google.com/mail/et.al.*
Instructor mail ID : *welcometanmay@gmail.com*

Technology Used ::

Programming Language : PYTHON3
Environment/IDE : JUPYTER NOTEBOOK
Package/Library : PANDAS(pandas), SCIKIT-LEANRN(sklearn),
SciPy(scipy), MATPLOTLIB(matplotlib)
NLTK(nltk), WORDNET(nltk corpus wordnet),

Thanks!

References

[1] A. K. Jain, R. P. W. Duin, and J. Mao, *Statistical pattern recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000

[2] Jerome H. Friedman, Robert Tibshirani, and Trevor Hastie., *The Elements of Statistical Learning*. Springer, second edition, 2008.

[3] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. *A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 226–231, 1996.

[4] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*, McGraw Hill, 1983.

[5] A. Hotho, A. Nurnberger, and G. Paa, *A brief survey of text mining. LDV Forum - GLDV Journal for Computational Linguistics and Language Technology*, 20(1):19–62, 2005.

[6] Python3, *PYTHON*, *https* : *//www.python.org*;

[7] Scikit-Learn, *sklearn*, *https* : *//scikit − learn.org*;

[8] Pandas, *pandas*, *https* : *//pandas.pydata.org*;

[9] SciPy, *scipy*, *https* : *//docs.scipy.org/* ; *and*
https : *//docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html*
;

[10] NLTK, *nltk*, *https* : *//www.nltk.org/*;

[11] WORDNET, *WordNet*, *https* : *//wordnet.princeton.edu/*; *and*
wordnetweb.princeton.edu/perl/webwn ;

[12] *WordNet*, *https* : *//en.wikipedia.org/wiki/WordNet*;

[13] *Morphological derivation https* : *//en.wikipedia.org/wiki/Morphological_derivation*

[14] *Lemma (morphology)*, *https* : *//en.wikipedia.org/wiki/Lemma_(morphology)*

All the wab-page links visit dated by April 28, 2019.