EE219 Project 4

Clustering

Winter 2017

Introduction:

Clustering algorithms are unsupervised methods for finding groups of data point that have similar representations in a proper space. Clustering differs from classification in that no a priori labeling (grouping) of the data points is available. As such, K-means clustering iteratively groups data points into regions characterized by a set of cluster centroids. Each data point is then assigned to the cluster with the nearest cluster centroid. In this project the goal is to find proper representations of the data and evaluate the performance of clustering algorithms.

Dataset:

We work with "20 Newsgroups" dataset that we already explored in project 2. It is a collection of approximately 20,000 documents, partitioned (nearly) evenly across 20 different newsgroups, each corresponding to a different topic. Each topic can be viewed as a "class".

In order to define the clustering task, we pretend as if the class labels are not available and aim to find groupings of the documents, where documents in each group are more similar to each other than to those in other group. These clusters capture the dependencies among the documents that are known through class labels. We then use class labels as ground truth to evaluate the performance of the clustering task.

In order to get started with a simple clustering task, we work with a well separable portion of data that we used in project 2, and see if we can retrieve the known classes. Namely, let us take all the documents in the following classes:

Class 1: Computer technology	**Class 2: Recreational activity**
comp.graphics **comp.os.ms-windows.misc** **comp.sys.ibm.pc.hardware** **comp.sys.mac.hardware**	**rec.autos** **rec.motorcycles** **rec.sport.baseball** **rec.sport.hockey**

We would like to evaluate how purely the a priori known classes can be reconstructed through clustering.

Problem Statement:

1) Finding a good representation of the data is fundamental to the task of clustering. Following the steps in project 2, transform the documents into TF-IDF vectors.

2) Apply K-means clustering with k=2. Compare the clustering results with the known class labels. Inspect the confusion matrix to evaluate how well your clusters match the ground truth labels. Is there a permutation of the rows that makes confusion matrix look almost diagonal?
In order to make a concrete comparison of different clustering results, there are various measures of purity a given partitioning of the data points with respect to the ground truth. The measures we examine in this project are homogeneity score, completeness score, adjusted rand score and the adjusted mutual info score. Homogeneity is a measure of how purely clusters contain only data points that belong to a single class. On the other hand, a clustering result satisfies completeness if all of its clusters contain only data points that belong to a single class. Both of these scores span between 0 and 1; where 1 stands for perfect clustering. The Rand Index is similar to accuracy measure, which computes similarity between the clustering labels and ground truth labels. This method counts all pairs of points that both fall either in the same cluster and the same class or in different clusters and different classes. Finally, adjusted mutual information score measures mutual information between the cluster label distribution and the ground truth label distributions.

3) As you observed, high dimensional sparse TF-IDF vectors do not yield a good clustering performance. In this part we try to find a "better" representation tailored to how the clustering algorithm works. Since in K-means clustering within-cluster distances are minimized in Euclidean l_2-norm sense, it turns out that we need to reduce the dimension of the data properly.
We will use Latent Semantic Indexing(LSI) and Non-negative Matrix Factorization(NMF) that you are already familiar with for dimensionality reduction. In order to get a good initial guess for an appropriate dimensionality to feed in the K-means algorithm, find the effective dimension of the data through inspection of the top singular values of the TF-IDF matrix and see how many of them are significant in reconstructing the matrix with the truncated SVD representation. You can then change the dimensionality and choose one that yields better results in terms of clustering purity metrics.

Now, use the following two methods for reducing the dimension of the data by sweeping over the dimension parameter in each. (You should work with values as low as 2-3 and up to the effective dimension you found above)

- Truncated SVD (LSI) / PCA
- NMF

Again, if the clustering purity is not satisfactory, try to find a better representation of the data by

- Normalizing features
- Applying some non-linear transformation on the data vectors [after reducing dimensionality].

To get a visual sense on the NMF embedding of the data, try applying NMF to the data matrix with ambient parameter 2 and plot the resulting points to choose the appropriate non-linear transformation. Can you justify why logarithm is a good candidate for your TFxIDF data?

Report the measures of purity introduced in part 2 for the best final data representation you use.

4) Visualize the performance of your clustering by projecting final data vectors onto 2 dimensions and color-coding the classes. Can you justify why a non-linear transform is useful?

5) In this part we want to examine how purely we can retrieve all the 20 original sub-class labels with clustering. Therefore, we need include all the documents and the corresponding terms in the data matrix and find proper representation through reducing the dimension of the TF-IDF representation. In doing so, try different effective ambient space dimension for both truncated SVD and NMF dimensionality reduction techniques and the different transformations of the obtained feature vectors as outlined above. Then, find appropriate parameter k to use in K-means clustering in order to find pure clusters with respect to the class labels

Report all purity measures described earlier for the final representation that you find for your data.

6) Evaluate the performance of your clustering in retrieving the topic-wise classes. Note that again, you need to find a proper representation of your data through dimensionality reduction and feature transformation.

Class 1: Computer technology comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	**Class 2: Recreational activity** rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	**Class 3: Science** sci.crypt sci.electronics sci.med sci.space
Class 4: Miscellaneous misc.forsale	**Class 5: Politics** talk.politics.misc talk.politics.guns talk.politics.mideast	**Class 6: Religion** talk.religion.misc alt.atheism soc.religion.christian

Submission: Please submit a zip file containing your report, and your codes with a readme file on how to run your code to ee219.winter2017@gmail.com. The zip file should be named as "Project2_UID1_UID2_..._UIDn.zip" where UIDx are student ID numbers of the team members. If you had any questions you can send an email to the same address.