ML hw4 物理三 王昱翔

- Analyze the most common words in the cluster.

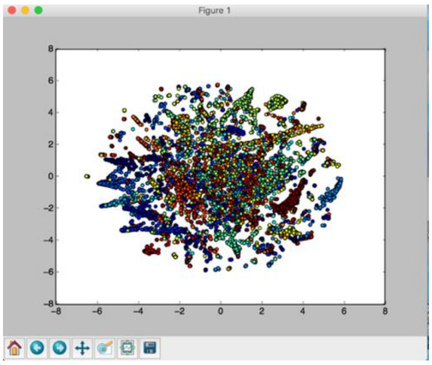
  Each common words in 20 categories are shown below.

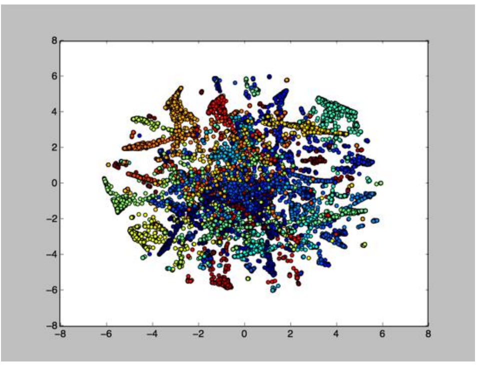
Categories	Most common word before(freq), after(freq) remove stop words
1	wordpress (697), wordpress(697)
2	oracle (648), oracle (648)
3	svn (460), svn (460)
4	apache (495), apache (495)
5	excel (751), excel (751)
6	matlab (575), matlab (575)
7	visual (670), visual (670)
8	a (336), cocoa(207)
9	mac (383), mac(383)
10	bash (512), bash(512)
11	spring (687), spring(687)
12	hibernate (667), hibernate(667)
13	scala (575), scala(575)
14	sharepoint (642), sharepoint (642)
15	ajax (608), ajax (608)
16	qt (457), qt (457)
17	drupal (642), drupal (642)
18	linq (712), linq (712)
19	haskell (490), haskell (490)
20	magento (732), magento (732)

  These common words analysis are based on labeled data. If we remove stopwords, each common words in each category are the same of the words of the real tags.

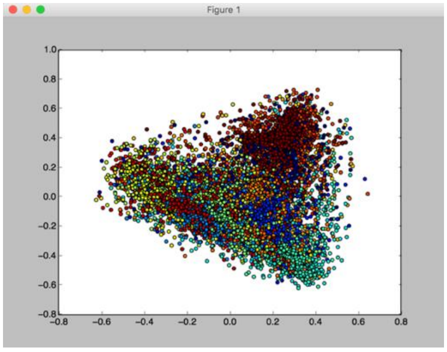
- Visualize the data

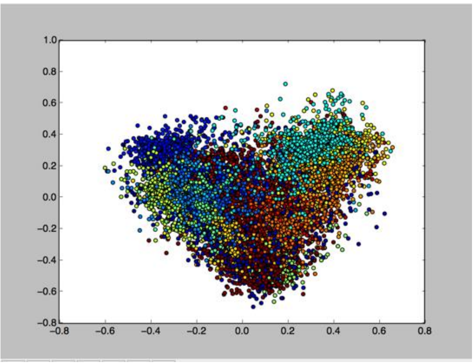
  Left side: True Labels Right side: Predicted Labels





t-SNE





PCA

In t-SNE and PCA, I reduce dimension based on the vector results from LSA. The most obvious difference is in the center of the t-SNE. There are a lot of different categories in the real labels figure, while there are few categories in the prediction labels figure. The reason that the left and the right figures are pretty similar is because the performance of LSA is almost about 0.9. In addition, we can find that t-SNE figure is sparser than PCA.

- Compare different feature extraction methods.

  I have come up with 4 methods, which are tf-idf, LSA, word vector, and Paragraph vectors respectively.

  1. Tf-idf

     Use Tf-idf and PCA reduce dimension from (≈12000 -> 24)

  2. LSA on Tf-idf

     I first used tf-idf and the dimension of TruncatedSVD is 24. Then I cluster my data via Kmeans and inverse transform to the center of the cluster in order to find the most common words in every cluster center. Let's call the common word tags W_{tag} and tag (range: 1-20) the sentences if sentences contain words in W_{tag} else labeled 0. During the testing we'll classify two sentences the same topic only when their labels are not 0 and the same.

  3. Word Vector

     I trained the word vector with the toolkit word2vec. The preprocessing is showed below, and the parameter of the word2vec (min_count=10, dim=400, window=15, sample=1e-3)

Corpus = title_Stackoverflow.txt + doc.txt

- Remove none English character
- Remove http or https link
- Lower all the letter
- Remove the stopwords (from nltk) and use word segment toolkit

After finish training the word vector, I cluster the sentence with the average embedding $W_{avg} = \quad W_{Total}$/len(Sent) via Kmeans. The performance will show in Table1.

4. Paragraph Vector

   I trained Paragraph Vector with Doc2Vec. However, the performance was very awful and I'm sure that I didn't make mistakes when input the sentences. I trained my vector both on the "title_Stackoverflow.txt" and "title_Stackoverflow.txt + doc.txt" and cluster via Kmeans. The parameter of Doc2Vec(dim=300, window=10, min_count=5, iter=10) .

Method	Tf-idf	LSA	Word2Vec	Doc2Vec
Kaggle score	0.257	0.878	0.592	0.124

Table1: Four Methods performance

- Try different cluster numbers and compare them.

Cluster number	15	18	19	20	21	22	25
Kaggle score	0.843	0.900	0.873	0.878	0.827	0.775	0.765

These results are all based on 24-dim LSA method.

We can easily know that why 18 cluster number have the best performance. According to LSA result, when the cluster number=20, the 20 tags are [u'hibernate', u'use', u'wordpress', u'qt', u'svn', u'drupal', u'excel', u'ajax', u'bash', u'magento', u'scala', u'apache', u'oracle', u'error', u'matlab', u'haskell', u'linq', u'spring', u'mac', u'sharepoint'], which showed that "visual" and the "cocoa" sentences can't be classified and some of the sentences might classified to the wrong tags such as "use" and "error", which are not the correct tags.

If the cluster number=18, these 18 tags are all the correct tags (no "use" and "error" compared with the above 20 tags). Therefore, LSA reach the best performance on kaggle if the cluster number=18