Master Data Science/MVA data competition 2017

Email recipient recommendation

Group : ZHANG (zz10086: Zhao ZHANG, xiyu.zhang : Xiyu ZHANG)
Sat 28 Jan 2017 – Sun 12 Mar 2017
Professor: Michalis Vazirgiannis, Antoine Tixier

After joining training info and training set by mid, we generate only one data frame for training. There are five columns: sender, who sent this message; mid, unique id of mails; date, the time stamp when sender sent this mail; body, the content of mail; recipients, the mail address of recipients. We processed test data set as the same way.

The goal is to send right e-mail to appropriate recipients. To improve baseline method, we added the analysis of body and date.

1. Feature Engineering

There are a lot of information in the body. For example, I can imagine if I was a traveler, I am willing to receive the information of flights or TGV; if I was a customer of Auchan, I would like to receive the promotion information. Further, in the body there consists the cc and senders' mail addresses. This is also a very important information. We think that based on the body, we can get some semantic characters for each body. Therefore, we used two methods as followed to extract features from bodies.

There are also time stamps to deal with for each mail. If a recipient received a mail at 4:am in the morning, it's reasonable to consider this mail as a spam. In addition, we found that the minimum date started on 2001-11-02 in test data set and the maximum date terminated on 2001-11-01 19:12:34 in training. So it is not necessary for us to consider "year" features. We merely stay month, day, hour, minute and put them in front of body to generate plained-text features. Next, we removed stop words, punctuations, continues blanks etc from content of mail.

1.1 Word2vec

Compared with tf-idf, word2vec uses more advanced vector representations of term. It's based on the distributional hypothesis, we can determine the meaning of a word by looking at its context. If two words occur in a same "position" in two sentences, they are very related either in semantics or in syntactics. So we transformed every sentence into one matrix and then convert this matrix to one vector of 300 dimensions by averaging each column. Besides, we can also catenate vectors to one long vector with fixed size by padding sentences(content of mail). This method is usually used as the first step of RNN or CNN for NLP problems. In our experiment, we chose the first method

rather than padding sentences. Of course, this is not better than padding sentences. However, calculating the cosine similarity of sentences becomes fast due to the short dimension. In fact, there exists another expensive metric Earth Mover's Distance. In the training procedure, we run the 20% of the dataset with 6 hours in intel core i5 cpu. It's too slow. So we give it up.

1.2 Tf-idf

After trying the word2vec, we tried use tf-idf to get a feature vector for each document under the guides of a paper[1]. In tf-idf, two vectors are used to calculate the "relationship" between their corresponding words. But this relationship is not semantically or syntactically related, it's just about the level of common occurrence in the textual units to be learnt from. Besides, we dummied the variable sender and put them behind tf-idf vector to get one new representation of item.

Then we tried PCA for dense matrices and SVD for sparse matrices to reduce the dimension of features . As to SVD, we change matrices from 20000+ dimension to fixed 1000 dimension. So that we got a better result with an acceptable time cost.

2. Model Tuning and Comparison

We never did a multiple labels classification with 9000+ labels. We think that since there are som many labels, the traditional supervised or unsupervised models will not have a high accuracy. And with our equipments, the calculating speed is too slow. So we used a simple "union" method to get final results. We used two methods: K nearest neighbor.

2.1 Model: KNN

Once we get one test item, we are going to calculate top 30 of nearest neighbors in our training data set and then make an union with the set of mids classified by sender. After that, we adopt the same idea with baseline. That is to say, we only regard the top 10 of recipients by frequency as predicted recipients.

2.2 Tuning

Here we divided the training set into two parts: training and valid. Technically speaking, because each sender appears in the test set, we have to sample randomly the training data set by sender. In this way, we could know K = 30 is not a bad parameter for KNN.

2.3 Comparison

When we used Word2vec, the metric of measuring similarity is cosine similarity. However, we made use of euclidean distance for tf-idf features in our experiment even if we should use cosine

similarity. As to the generalisation of our model, what we need to do is sampling differently the training data set many times by sender and doing k-cross validation to choose better parameters.

3. Conclusion

This project gives us a new version of machine-learning. In this project, we learnt a lot of methods that we never used before(e.g. Earth Mover's Distance) [2]. And we know that to deal with the text is a very difficult work.

The data is well formed and important. We can use these data to apply to many different interesting areas like medicine, health-care industry.

References:

[1]: Ramnath Balasubramanyan, Vitor R. Carvalho, William Cohen(2008), CutOnce - Recipient Recommendation and Leak Detection in Action

[2]: Kusner, M., Sun, Y., Kolkin, N., & Weinberger, K. Q. (2015). From Word Embeddings To Document Distances.