Textural Query Image Retrieval System

December 13, 2019

Abstract

In this paper, we addressed the task of retrieving the most relevant images in an image set given a natural language query. We have trained and experimented with 3 different models to accomplish this challenge, each with varying results. We employed Support Vector Machine + K-nearest Neighbor, Partial Least Squares Regression and Ridge Regression. Finally, the best performing algorithm was the Ridge Regression, which achieved an accuracy of 0.3896.

1 Introduction

For this competition, our goal is to investigate different approaches to build a large-scale image search engine. We are giving a testing image set, training image set, text description made up of five sentences, feature vectors for each image and tags for both training and testing image.

Basically, our approaches are using textual based algorithms to train the input natural language descriptions and find images in training set with corresponding description, which includes SVM, PLSR and Ridge Regression. SVM classifier can perform well on high dimensional data and solve unbalanced data distribution.Olivier2008PLSR is good for modeling a response variable when there exist many predictor variables that are highly correlated or even col-linear. Ridge regression is good at reducing standard errors and preventing the model from over fitting.

In this paper, the three models we mentioned above are all described and elaborated in detail, as well as how we map text descriptions to ResNet features and obtain a set of images that matches best.

2 Data Description and Preprocessing

2.1 Data Description

The training and testing data is downloaded from Kaggle website:

`https://www.kaggle.com/c/cs5785-fall19-final/data`

The training data has 10000 JPG image of size 224*224.
The testing data has 2000 JPG image of size 224*224
The data sets contain: natural language descriptions, images, ResNet features (fc1000 and pool5) and tags.

2.2 Evaluation

For each description, students are required to submit 20 candidates, and the system will evaluate the results using the MAP@20 metric, returning the score based on the ranking of the correct image.

2.3 Bag of Word (BoW) Preprocessing

The initial preprocess method used natural language processing package to lowercase letters, lemmatization all words, and eliminate stop words and punctuation. After obtaining the initial bag of words set, we noticed that the number of unique individual words is around 6700. This high dimensional features can generate noise and increase our models' complexity.

Then we count frequency for each unique word and rank them. By observing the top 1000, 1500, and 2000, we found that occurrences for the last word 'downhill'(1000 case), 'hike'(1500 case), 'avocado'(2000 case) are 25, 12, and 7. In addition, either 'hike' or 'avocado' took a very small percentage in both tags and description. Thus, taking consideration of trade-off between noise and losing information, we decided to use the top 1000 frequent words as our bag of word set.

3 Experiment

3.1 Support Vector Machine

Support Vector Machine is a linear classifier that solves the following optimization problem:

$$min_{w,b,\zeta} \frac{1}{2} w^T w + c \sum_{i=1}^{n} \zeta_i$$

subject to

$$y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i$$
$$\zeta_i \geq 0, i = 1, ..., n$$

We choose to use descriptions as input and employ the SVMs to achieve predictions in tags. The outcome has to be binary, which means that it can only have two outcomes. For our model, our outcome is either "0" or "1" which implies that whether a tag is associated with a description or not. Furthermore, we decide to use K-Nearest Neighbor to find the Top 20 relevant image, which is a method used for classification and regression. If k = 1, then the object is simply assigned to the class of that single nearest neighbor. We employ SVM on the BoW dataset and get an accuracy of 0.15762.

3.2 Partial Least Squares Regression

Our second method applies Partial Least Square Regression(PLSR) to predict testing image feature vectors. PLSR model combing wrapper feature selection can improve classification accuracy (Tian et al., 2012). This process enlightens us to use the PLSR to identify image description vectors. Through our BOW preprocessing, we can generate training description BOW vectors and actual testing description BOW vectors.

The PLSR uses training ResNet features(pool-5) set as input which contains more information than the fc1000 layer and predicted description BOW vectors as output, to calculate coefficient matrix. To determine the number of components of our model, we applied cross-validation with 5 folds, and tested 50, 100, 150, 200 components for training data. Then we picked the model with 100 components

with the highest average accuracy.

Using our model on testing ResNet features set, we can generate predicted testing BOW vectors. Since our search engine needs to get top 20 related pictures, we studied the similarity between each actual testing description BOW vector and all predicted testing BOW vectors. Patterns with great different attribute values may have a high similarity measure (Xia et al., 2015), so instead of using euclidean distance, we decided to use cosine similarity and recorded the top twenty similar vectors from predicted testing bow vectors.

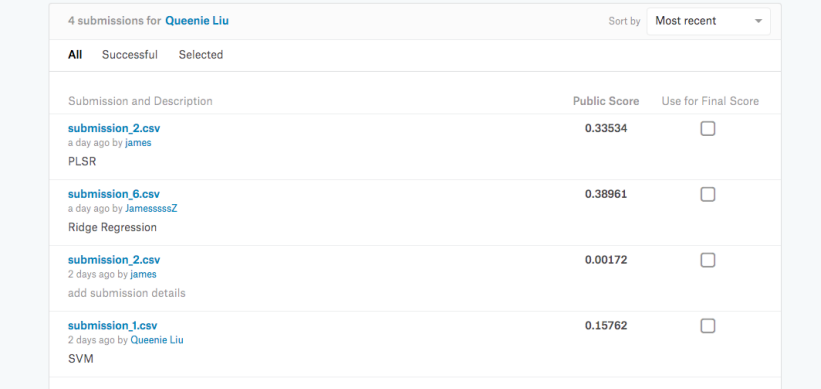
3.3 Ridge Regression

Our third method is Ridge regression. Ridge regression is a technique particularly used for analyzing the multicollinearity in linear regression.

$$\sum_{i=1}^{M}(y_i - y_i^*) = \sum_{i=1}^{M}((y_i - \sum_{j=0}^{p} w_j * x_{ij})^2 + \lambda \sum_{j=0}^{p} w_j^2$$

The cost function of ridge regression is similar to linear regression. However, the ridge regression contains an extra term (λ) as penalty, which can regularize the coefficients so that if the coefficients take large values the optimization function is penalized. Compared with linear regression, ridge regression can reduce model complexity and prevent over-fitting.Saptashwa 2018
Our Ridge model also uses training ResNet features set (pool5) as input and description BOW vectors as output. We get 0.38961 accuracy for our ridge model which is also the best performance we got.

4 Results



According to the performances at Kaggle, we find that the best performing algorithm is Ridge Regression which achieved an accuracy of 0.38961. Since Ridge Regression contains extra penalty which prevents over-fitting and the PLSR does not have such feature, the testing performance for Ridge Regression is better.

5 Conclusions

In this competition, we tried multiple ways to define and address the image retrieval problem. By preprocessing raw data and constructing different scripts, we end up with getting three models to map and retrieve related images.

According to the accuracy chart above, it is obvious to say that the efficacy of using Ridge Regression model is the best among all three models, which reaches almost 40 percent accuracy rate. For the SVMs + KNN model, the mapping accuracy is worse than expected and there is no highly correlated relationship between images and natural language descriptions. For PLSR, a linear regression model in the projection space, it performs almost as good as Ridge Regression with 34 percent in accuracy rate.

6 Citations, references

References

[1] Tian, Wen-Meng, et al. *Key Process Variable Identification for Quality Classification Based on PLSR Model and Wrapper Feature Selection.* Proceedings of 2012 3rd International Asia Conference on Industrial Engineering and Management Innovation (IEMI2012), 2012, pp. 263–270., doi:10.1007/978-3-642-33012-427

[2] Xia, Peipei, et al. *Learning Similarity with Cosine Similarity Ensemble.* Information Sciences, vol. 307, 20 Feb. 2015, pp. 39–52., doi:10.1016/j.ins.2015.02.024.

[3] Randall D. Tobias. *An Introduction to Partial Least Squares Regression.* SAS Institute Inc., Cary, NC, 2016.

[4] Olivier Chapelle, S. Sathiya Keerthi†. *Multi-Class Feature Selection with Support Vector Machines.* 2008.

[5] Saptashwa Bhattacharyya. *Ridge and Lasso Regression: A Complete Guide with Python Scikit-Learn.* Towards Data Science, 2018