Ying Zhou(yxz179830), Yankai Jia(yxj171130), *University of Texas at Dallas, CS 6375 Machine Learning Project*

Face classification

***Abstract*—This paper classifies human face by both SVM and K-means models, evaluates this two models based on the result. The data we use is from Ryan White, Ashley Eden, Michael Maire, "Automatic Prediction of Human Attractiveness", CS 280 class report, December 2003.**

***Index Terms*—SVM, Face detection, K-means**

I. Introduction

The aim of the project was to develop face classifier. There are many features which faces may be classified, e.g. nose, mouth, eye, ear, etc. People can classify faces as beautiful and ugly by identify the combination of different features and give their score for different by their standards. We classify the data set and want to know the relation of different features and score and we want to classify the gender by these features.
In order to make the problem simple, we considered five features: right eye, left eye, nose, right corner of mouth, left corner of mouth.

A. k-means algorithm

K-means which is an unsupervised learning method is popular for cluster analysis. The idea of k-means is to partition n data into k different clusters. Each training data has the nearest means to its centroid, which means they have some similar attributes. By using k-means we can cluster all training data(both female and male) by five features and analysis the relation between the score and features.

B.SVM algorithm

Support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.

II. Data Collection

A data set containing score-based features values from face detector and rectified corresponding to each image was obtained from Ryan White, Ashley Eden, Michael Maire, "Automatic Prediction of Human Attractiveness". This dataset contained scores which had ratings between 1-10 was used for binary gender classification and appearance classification. For gender classification, we combined image file for both female and male and we extracted 2/3 of the dataset as training data and the rest 1/3 of the dataset as test data for SVM.

III. Data Classification From Features and Score

The dataset we obtained had a default dictionary containing five different features(right eye, left eye, nose, right corner of mouth, left corner of mouth) that captured from the image of Hotornot website' s users and the score for each image that determined by the users of Hotornot websites. This data set had around 2250 samples.

IV. Tools and Resources

LIBSVM is an integrated software for support vector classification, (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR) and distribution estimation (one-class SVM). It supports multi-class classification. We use libsvm to do the SVM model on our face classification data. Download the zip file lib-svm-3.22 from https://www.csie.ntu.edu.tw/~cjlin/libsvm/ and install the svmutill library to the path of python installed on the computer.

V. Model Evaluation

A. k-means algorithm

For unsupervised k-means clustering to work on our data set, we need to know which k value can better represent our data set also we need to determine how to represent cluster centroids and how to update to better centroids each iteration.

To solve this, we chose linear distribution, and initialized the two and three centroids and calculated the distances between these centroids and each data (as determined by Euclidean distance). Once a group of images is assigned to a centroid, the centroid is updated by its corresponding the mean of the distance matrices of those samples therefore we can get a new centroid and continue doing this until the centroids did not change.

For k=3, we get figure 1.

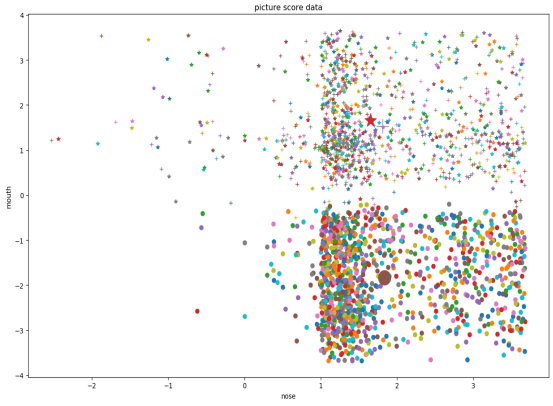

Figure 1.K-means when k=3

For k=2, we get figure 2.

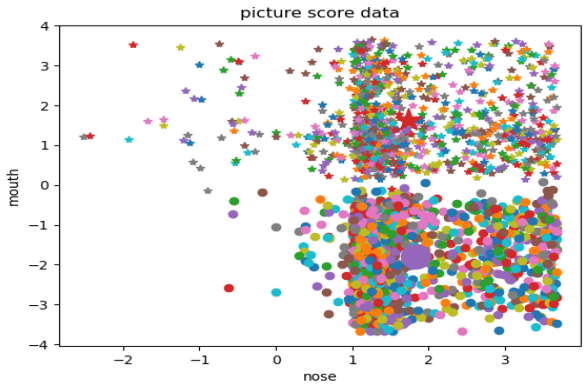

Figure 2.K-means when k=2

By comparing these two results, we can get that when k=3, the class centroids represented by + and the class centroids represented by * are very close. By analyzing k=2, we can get the result that all training data are clearly divided into two parts. Therefore, we choose k=2 to analyze the relation between score and features.

B.SVM algorithm

Support Vector Machine (SVM) is a supervised discriminative classifier defined by a separating hyperplane. The idea is to find the largest margin to divide the data set into two parts which each part means a class. However, if the margin is too large, the classifier will be overfitting and data set may be not linear. Therefore, we need to find a correct kernel function to represent the SVM algorithm.

To solve this, we have tried three different kernel functions which are linear kernel function, polynomial kernel function and sigmoid kernel function. By defining its corresponding dual problem and get the value of each parameter for the $y = \omega x + b$, we can get the classification for the test data and get the accuracy.

For linear kernel function, we get the result as figure 3.

```
>>> import numpy
>>> import os
>>> os.chdir('C:\libsvm-3.22\python')
>>> import sys
>>> sys.path.append('C:\libsvm-3.22\python')
>>> from svmutil import *
>>> y,x=svm_read_problem('train.txt')
>>> yt,xt=svm_read_problem('test.txt')
>>> model=svm_train(y,x)
>>> p_label, p_acc, p_val = svm_predict(yt, xt, model)
Accuracy = 60.7435% (817/1345) (classification)
>>> print(p_label)

>>>
```

Figure 3. Code of linear kernel SVM

For polynomial kernel function, we get the result as figure 4.

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Inte
l)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import numpy
>>> import os
>>> os.chdir('C:\libsvm-3.22\python')
>>> import sys
>>> sys.path.append('C:\libsvm-3.22\python')
>>> from svmutil import *
>>> y,x=svm_read_problem('train.txt')
>>> yt,xt=svm_read_problem('test.txt')
>>> model=svm_train(y,x,['-t',1])
>>>  p_label, p_acc, p_val = svm_predict(yt, xt, model)
SyntaxError: unexpected indent
>>> p_label, p_acc, p_val = svm_predict(yt, xt, model)
Accuracy = 60.3717% (812/1345) (classification)
>>>
```

Figure 4. Code of polynomial kernel SVM

For Sigmoid kernel function, we get the result as figure 5.

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 3
l)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import numpy
>>> import os
>>> os.chdir('C:\libsvm-3.22\python')
>>> import sys
>>> sys.path.append('C:\libsvm-3.22\python')
>>> from svmutil import *
>>> y,x=svm_read_problem('train.txt')
>>> yt,xt=svm_read_problem('test.txt')
>>> model=svm_train(y,x,['-t',3])
>>> p_label, p_acc, p_val = svm_predict(yt, xt, model)
Accuracy = 46.7658% (629/1345) (classification)
>>>
```

Figure 5. Code of Sigmoid kernel SVM

By comparing these three results, we can get that when using linear kernel function, the accuracy is higher. Therefore, we choose linear kernel function to analyze the classification of gender as table 1.

Kernel function	Accuracy
Linear function	60.74%
Polynomial function	60.37%
Sigmoid function	46.76%

Table1

VI. ANALYSIS

A.k-means algorithm

By using k-means algorithm, we can compare the relation between score and different features.

For female, we compared the left eye and nose as figure 6.

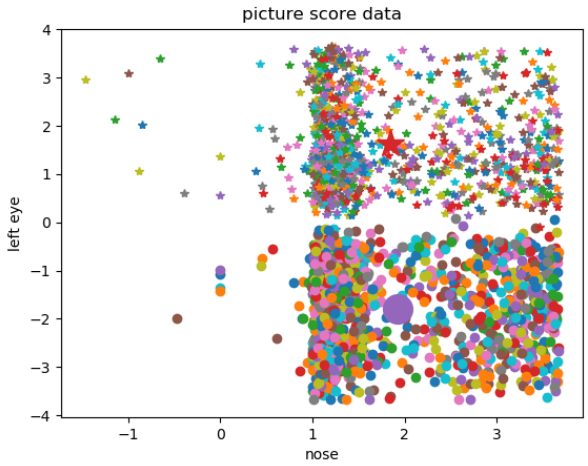

Figure 6

From figure 6, we can get the result that the data set are mainly divided into two parts. However, there are some noise data represented by * and o. Also, the data represented by * is sparse in the middle which can not represent precisely. In addition, by analyzing the figure, we can get the result that people have a higher weight on nose than eye.

Also, we compared the relation between right corner of mouth and nose(figure7), we cannot clearly classify the data set into two parts because the data are combined with each other although they have different classes.

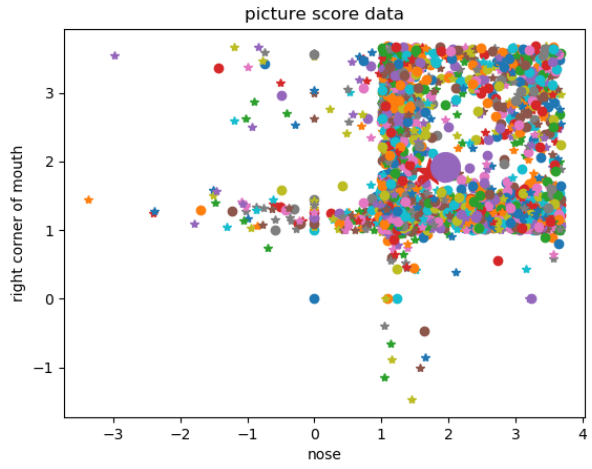

Figure 7

For female, the test the accuracy of cluster, we divided the score which range from 1 to 10 into 2 parts. One part is the score bigger than 6 and other one is the score lower than 6. If the score is bigger than 6, we identify it as good and if the score is lower than 6, we identify it as bad. Then we combined score with these five features, because data set is already classified into two parts, we wanted to know the accuracy of clustering. By comparing with k=2 and k=3 we got the result as table 2.

k	2	3
Accuracy	0.5588104	0.4571682

Table 2

From the table2, we can also get the result that when k=2, the accuracy is higher than when k=3. So that it can classify data more correct.

B.SVM algorithm

By using SVM algorithm, we can classify the data set into two parts by gender (female or male) by five different features that captured by face detector. By running the program, we get the accuracy of the classification is 60.76% by using polynomial kernel function. In addition, we want to know the reason why the accuracy rate is not high, we extract 1/3 and 2/3 and 1/2 of the training data, then we compute the accuracy of these new training data and we found that when the training data become smaller, the accuracy become lower. Then we found one of the reasons that caused the accuracy low is because the training data is not enough. Accuracies for different training data are showed as table 3.

dataset	Accuracy
1/3	50.77%
2/3	57.32%
1/2	53.18%

Table 3

VII. CONCLUSION

Both support vector machine and k-means are good algorithm for classification and they performed well. However, there are some cons for each algorithm.

For k-means, the initial centroids are very important because further computations are based on these initial centroids. Also, the value of k is important. in our previous computation, the accuracy when k=3 is not as good as when k=2. Also, the influence of noise data is another main reason. When k=2 and we want to represent the relation between left eye and nose, there are some noise data which may hurt the accuracy. In addition, k-means may stop by local optimum, however we want to get global optimum. Like figure 2, data with different classes are combined.

For support vector machine, we need to preprocess our training data because it may contain noise data. Also, we need to use cross-validation.

Also, the standards for people to tell a face is beautiful or not are different. For the same face, some people may think that is good and other people may have different ideas. Therefore the data set itself have errors.