A Comparison between Approaches of Classification for Recognition with Handwriting Digit Data

Fang Linjiajie(20382284), Xiao Ziliang(20378386), Xue Zexiao(20403674), Zhang Wenyong(20377289)

Introduction

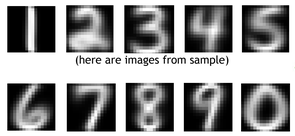
There exists many methods for classification in machine learning such as SVM, LDA, random forest, bagging, boosting…In this project, we will make classifications with bagging, random forest and SVM. For the response variable in this data are integers, we will also use logistic regression to predict the input handwriting digits matrices. During the training process, we do parameter selection for random forest model so that we can choose a model with a certain tree number. Through 5-fold method, we divide dataset into five parts with equal size. and make cross validation with four parts and obtain the model with lowest training error. Finally, we input test data and calculate the test score expressed by scores and make sure which model have relatively better behavior.

Background

Nowadays, image recognition is increasingly popular a lot of handwritten numbers need to be recognized correctly by computer rather than by person, which can make contribution to enhancing the efficiency of our society. In this project, our purpose is to choose a proper classification model to make this recognition, that is, input an image, output the corresponding number.

Handwriting digit data

Just as last mini project, we also choose handwriting digit data in this project. This data is composed by 7291 handwritten numbers from 0 to 9 which are collected from postoffice in thee city. Every number is written by a postman and output as a 16 by 16 gray matrix after scanned by the computer. Therefore, in order to make right classification most precisely, we use this data training four different model as mentioned before and calculate test error of each model and finally choose the model with the lowest error to become the “champion model”.



(here are images from sample)

Methodology

➢ Bagging

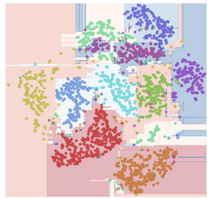
Algorithm of bagging

Bagging means bootstrap and aggregation, mainly reducing the variance but not bias. Bagging trees works well because trees seems with high variance and low bias. For regression, we get our training data set: $Z = z_1, z_2, \ldots, z_n$, where $z_i = (x_i, y_i)$. And construct B bootstrap sample sets each of them is random picked from the training data set with same size. Then fit the model with B replications and observe $\hat{f}(x)^b$ where $b = 1, 2, \ldots, B$. The bagging estimate is defined by

$$\hat{f}_{bagging}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^b(x)$$

Because this method simply calculates average, so the method can just reduce variance but not bias. For classification, let $\hat{C}^b(x)$ be the classification for the b_{th} sample. Then $\hat{C}(x)$ represent the total classification will be the majority of the $\hat{C}^b(x)$. [1]

Result



➢ Random forest classifier

Algorithm of random forest

1. For $b = 1$ to B:
   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.
   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. Select m variables at random from the p variables.
      ii. Pick the best variable/split-point among the m.
      iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression: $\hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$.

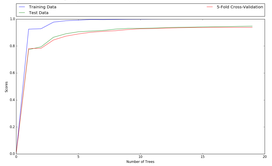
Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{rf}^B(x) = majority\ vote\ \{\hat{C}_b(x)\}_1^B$.

So why we use random forest?

From the procedure of first step, we draw a bootstrap sample in each loop, we can efficiently reduce the variance of bagging by reducing the correlation between the trees, without increasing the variance too much. This is achieved in the tree growing process through random selection of the input variables.

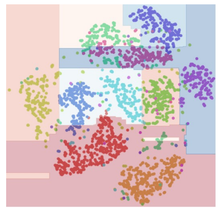
Parameter Selection-“Random Forest”

For further exploration, we choose random forest as our pre-adjusted model to choose a optimal pruning parameter. Since the complexity of our forest can significantly influence the models’ ability of leaning. So we choose the number of trees in [1,20] as our pruning parameter selection pool(while the default number is 10 in python). Our selection is shown as belc



From this process, we find that the optimal number of trees is 16. Because it can give a model with the lowest training error when we do the cross validation.

Result



➢ Support Vector Classifier

Why we choose SVC as a candidate?

1.It has a model with parameter, which make the user think about avoiding overfitting.
2.It has kernel function, one can adjust kernel when meeting different boundary of classes.
3.From a human’s intuition, the difference between shapes of numbers are relatively clear.

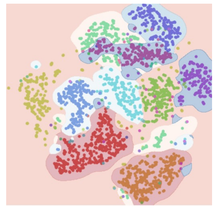
The model and the solution function can be written as below:

$$L_D = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{i'=1}^{N}\alpha_i\alpha_{i'}y_iy_{i'}\langle h(x_i), h(x_{i'})\rangle.$$

$$\hat{f}(x) = \sum_{i=1}^{N}\hat{\alpha}_i y_i K(x, x_i) + \hat{\beta}_0.$$

After several try with different kernel functions, we finally decide just to use linear kernel function. Although the training error of th model with linear function is the lowest, we can intuitively think that one can hardly regard six as seven because the threshold between two digits is clear and such classifier can also be estimated by a hyperplane.

Result

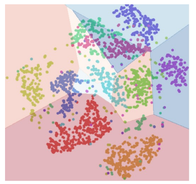


➢logistic Regression model

There are several key difference between SVM and logistic regression model:
1. Logistic model fits the data as if they are along a continuous function.
2.SVC could have difficulty when the class are not separable or there is not enough margin to fit a hyperplane between the two classes.
The raw model of logistic model with K targets is shown as below:

$$\log\frac{\Pr(G=1|X=x)}{\Pr(G=K|X=x)} = \beta_{10} + \beta_1^T x$$
$$\log\frac{\Pr(G=2|X=x)}{\Pr(G=K|X=x)} = \beta_{20} + \beta_2^T x$$
$$\vdots$$
$$\log\frac{\Pr(G=K-1|X=x)}{\Pr(G=K|X=x)} = \beta_{(K-1)0} + \beta_{K-1}^T x.$$

Result



Prediction and Conclusion

Training score and test score can be seen in the following form.

Model	Bagging	Random forest (Est.=16)	SVC	Logistic
Training Score	99.62%	99.97%	98.73%	99.18%
5-fold-cv Score	89.52%	93.91%	96,43%	94.05%
Testing Score	91.30%	93.80%	96.63%	93.66%

We can draw conclusions that:
1. Random forest can make a better classification than bagging. The former can reduce correlation between trees thus the next pruning can be less dependent on last one.
2. SVC is better than logistic regression, for the boundary of SVM is sharper than logistic and more robust and test score is higher.
3. It is obvious that testing score is very close to cv score, that means 5-fold-cross validation have a good performance. We can also see training score behaves much better than 5-fold-cv score and test score, that satisfies our basic knowledge.
4.If we care about computational efficiency, bagging runs faster than random forest, logistic is faster than SVC.
5. From above four small conclusions, it is suffice to claim that **Support Vector Machine** win the competition because its testing score wins a lot than others. However, consider computational efficiency, **Random forest** is also not a bad choice because it runs faster than SVC and also have a good behavior.