## Name: He Zhiming S-ID: 2018227154 Date: 2019-11-26

**Introduction**

Handwritten digit recognition is a research hotspot with high practical value in image processing and pattern recognition. Handwritten numeral recognition is to allow a computer simulation person to automatically recognize handwritten Arabic numerals on paper. I have used 3 ML techniques in *MNIST Handwritten Data*: **1).** Random Forest (Supervisory Algorithms), **2).** Support Vector Machine (Supervisory Algorithms), **3).** K-means (Unsupervised Algorithms). Due to insufficient performance of my personal computer, I chose Google's online open source IDE: **Colab**. Therefore, all my code files need to be uploaded to the Colab website for verification. In addition, it has been tested that if you install Pycharm software locally, you can also view the source code of my project (whether other software is feasible and not practiced).

**Method**

**1).** Random Forest

In Machine Learning, Random Forest is a classifier that contains multiple decision trees, and the category of its output is determined by the mode of the category of the individual tree output. The main principles of the Random Forest algorithm are **a).** Suppose ***N*** is the number of training cases (samples), and ***M*** is the number of features. **b).** The number of input features ***m*** is used to determine the decision result of a node on the decision tree; where ***m*** should be much smaller than ***M***. **c).** Sampling ***N*** times from the ***N*** training use cases (samples) with a sampling method to form a training set (i.e., bootstrap sampling), and use the unselected use cases (samples) to make predictions and evaluate their errors. **d).** For each node, m features are randomly selected, and each node's decision on the decision tree is determined based on these features. Based on these m features, calculate the best splitting method. **e).** Every tree will grow completely without pruning, which may be used after building a normal tree classifier.

**2).** Support Vector Machine

Support Vector Machine (SVM) is a kind of generalized linear classifier which classifies the data according to the supervised learning method. Its decision boundary is the maximum-margin hyperplane for learning samples. SVM uses the hinge loss function to calculate the empirical risk and adds the regularization term to the solution system to optimize the structural risk. It is a classifier with sparsity and robustness. SVM can be classified by kernel method, which is one of the common kernel learning methods.

**3).** K-means

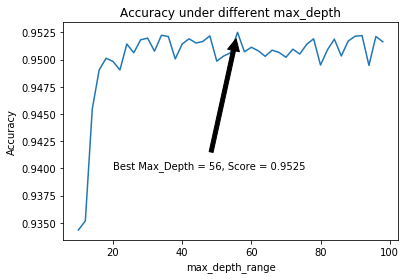
K-means clustering algorithm is an iterative clustering analysis algorithm. Its steps are to randomly select k objects as the initial clustering center, then calculate the distance between each object and each seed clustering center, and assign each object to the nearest clustering center. Cluster centers and objects assigned to them represent a cluster. Every time a sample is allocated, the cluster center will be recalculated according to the existing objects in the cluster. This process will be repeated until a termination condition is met. The termination condition can be any of the following: **a).** No (or minimum number) objects are reassigned to different clusters. **b).** No (or minimum number) cluster centers change again. **c).** The squared error and the local minimum.

**4).** Aid

**a).** **Scikit-learn (sklearn)** is a well-known and open source python machine learning package. It contains many top-level machine learning algorithms, which have six basic functions, namely classification, regression, clustering, data dimension, model selection, and data pre-processing. It greatly facilitates my use of algorithms to process my MNIST dataset. **b). Colaboratory** (also known as Colab) is a free Jupyter Notebook environment that runs in the cloud and stores notebooks on Google Drive. Colaboratory supports Python 2 and Python 3 kernels, and provides a powerful GPU and TPU runtime environment. The most important thing is that it is free.

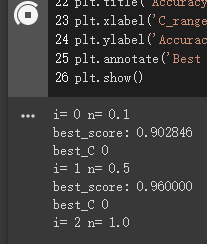
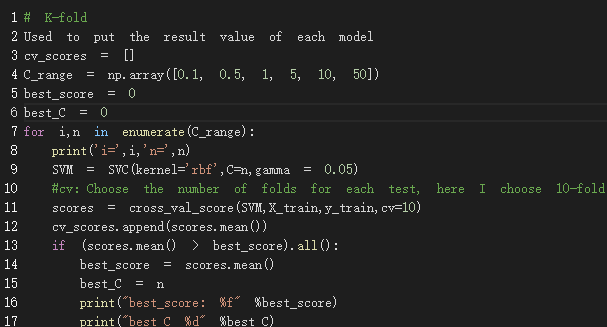
**Results**

**1).** Random Forest

Considering the **RandomForestClassifier()** function, **max\_depth** represents the depth of the decision tree. If it is equal to None, it means that the decision tree will not limit the depth of the subtree when constructing the optimal model. If the model has a large sample size and many features, it is recommended to limit the maximum depth; if the sample size is small or there are few features, the maximum depth is not limited. For the MNIST data set, exploring the effect of **max\_depth** on accuracy is an effective research direction! That's why I used the **cross\_val\_score()** function and used **10-Fold** cross-validation to verify the best value of **max\_depth** (range: 10 to 100, step size is 2). **(in MNIST-RandomForest-CV.ipynb file, the result graph is retained).** 

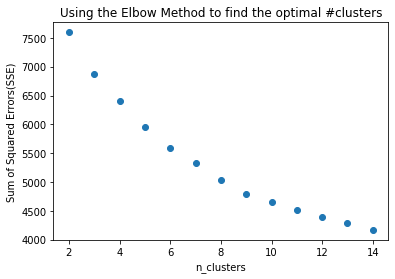
**2).** Support Vector Machine

When there are many training samples, SVM will run very slowly. So, when I run online, I always disconnect after waiting for the result, which makes it impossible for me to complete the parameter drawing of the super parameter **C** (**penalty factor**) temporarily. I will supplement this part when I present (use the **plt.plot** function for the parameter drawing of **C**). In addition, referring to the experimental results of other papers, I used the **Gaussian kernel function** and set **gamma** = 0.05 as the default condition, in order to obtain higher accuracy. I guarantee the accuracy of my code (because I have some screenshots of the results).



**3).** K-Means

Because the K-means algorithm itself has the following two defects: **a).** The **Sum of Squared Errors (SSE)** is actually assuming that the cluster is convex and of the same polarity (because it minimizes the distance from the center of mass), but the fact is not necessarily This is so, it responds poorly to elongated clusters, or manifolds with irregular shapes. The error rate of the clustering method is increased. **b).** The **SSE** in the high-dimensional eigenvalue data set, the formula for calculating Euclidean distance will expand rapidly. At this point, we need to use some methods to **dimensionality** **reduction**, and then use the K-means algorithm.

图片包含 文字, 地图, 天空

描述已自动生成So, I used Elbow Method to find the optimal value of **k** in K-Means. In addition, **t-SNE** (Stochastic Neighbors Embedding) is used to reduce the dimensionality of the dataset in order to draw re-delineated clusters in 2D scatter plots using matplotlib and **plotly**.

**Discussion**

I can clearly feel that the running time of the **SVM** is very long compared to the other two algorithms, and it is estimated that it can reach a gap of a hundred times, but the accuracy of the SVM is really high. When the model is running, I specifically set it to reduce the running time. **test\_size = 0.7** in the **train\_test\_split** function, but the accuracy can still reach **97%**, which is better than others.

**Conclusion**

Through this experiment, I fully understood and applied these three algorithms of machine learning. Although I just called the algorithms in the open source tool library, but through the explanation of the principles in the class, the comparison of the experimental results and the review of the data, I took me into the magical world of machine learning. Colab's ipynb file will keep all the code running results, **so the code I submitted retains the experimental results and result graphs.**

Machine learning is currently the hottest research direction. I believe I can also apply its technology to my research field (Fog Network Planning and Design). **Professor, thank you very much for taking me to the threshold of Big Data.**

Due to space limitations, I will mention **cross entropy** in my presentation.