

January Submission

Assignment

Name: Xu Zhang

Student number: 2200467

Major: Software Engineering and Artificial
Intelligence programme

Email: 2200467@student.uwtsd.ac.uk

Abstract- Machine learning is an important method and tool in current big data processing, and the use of machine learning to process image classification is a hot topic of research among scholars at present. Among them, K-nearest neighbor (KNN) and decision tree algorithm are simple, intuitive and theoretically mature, which are two relatively easy machine learning methods. At the same time, many new image datasets for image classification experimental training such as MNIST, have appeared on the web. In this paper, we validate the classification effect on top of Fashion MNIST image set by machine learning methods such as KNN, decision tree. A machine learning algorithm for image classification is performed using confusion matrix, F1-score, and accuracy rate as classification result evaluation metrics.

Keywords: Machine learning, KNN, Decision Tree Classifier, Fashion MNIST

I. INTRODUCTION

With the continuous development of Internet technology and the popularity of applications, various types of data including images, text, video, audio, signals, etc. are emerging. Machine learning is the mainstream method used to process these data, which is essentially a research computing method [1] that learns information about data features from the data and mines the corresponding models. Since the introduction of the world's first camera in the 1880s, images have occupied an important place in the history of human exploration of knowledge as one of the richest media for carrying information. As early as the 1970s, image classification appeared in people's lives in the form of text annotation. It was not until the 1990s that a new approach to image classification based on the information in the image itself was proposed, bringing the development of image classification techniques to a new frontier. In traditional machine learning, the standard process to recognize and classify an image is feature extraction, feature filtering, and finally inputting the feature vector into a suitable classifier to complete the classification. Automatic image classification is a key task in many application areas, including content-based image retrieval, Internet data filtering, medical applications, and industrial inspection based on machine vision. Applications of image technology, especially image recognition and classification, are rapidly developing in areas such as remote sensing, biometric recognition, and image detection.

In 2012, Alex Krizhevsky's breakthrough Alex Net network structure [2] won the 2012 ILSVRC (Image Net Large Scale Visual Recognition Challenge) with a record low error rate of 15.4%, this milestone historical event in the field of image recognition, blowing the horn of the explosion of machine learning in the computer field. The emergence of the Res Net model in 2015, with an ultra-low error rate of just

3.6%, also surpassed the average human recognition level, signaling the beginning of the rise of artificial intelligence in the human arena.

With the rapid development of the data era, image classification techniques have achieved rich research results in the directions of machine learning, pattern recognition, and data mining. A series of classification algorithms have been proposed, such as KNN (K-Nearest Neighbor) [3], NB (Naive Bayesian Model) [4], SVM (Support Vector Machines) [5], DT (Decision Tree) [6], and so on. Among them, KNN algorithm is a nonparametric classification method, and since it was proposed in 1968, its special characteristics [7] have made it widely researched and applied in the field of machine learning and computer vision. As one of the important algorithms in the field of data mining, it is more efficient and intuitive in class decision making with only a very small number of neighboring samples and can be used for classification without discriminating involving the class domain, so it is especially suitable for class domains consisting of overlapping and intersecting sets of samples. At the same time, decision trees are an early application to data mining classification problems, and in the face of large amounts of image data, decision tree algorithms can quickly build classifiers to achieve correct classification [8]. The decision tree algorithm contains a root node, a series of internal nodes and branches, and some leaf nodes.

The MNIST dataset is a subset of the NIST (National Institute of Standards and Technology) dataset. It consists of 60,000 training images and 10,000 test images. Each image is a 28*28-pixel point grayscale handwritten digital image of 0 ~ 9 with white characters on a black background [9]. Fashion MNIST dataset is the same as MNIST in that both are grayscale images, but Fashion MNIST dataset includes ten clothing images inside, and the number of training set and test set are also 60,000 and 10,000 [10].

In this paper, two algorithms, KNN and decision tree, are used to perform image classification experiments using Fashion MNIST as the dataset for the current popular fields of ML and image classification. The F1-score, confusion matrix and accuracy rate are used as evaluation metrics to check the classification effect of the algorithms. All experiments are run in python environment.

II. Related searches

A. Fashion MNIST

Fashion MNIST is a dataset of Zalando article images, which includes 10 categories of clothing images, namely: t-shirt, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, ankle boot.

Some of the Fashion MNIST images are shown in Fig. 1.



Fig. 1 Fashion MNIST images

B. KNN

KNN is a theoretically mature machine learning algorithm that can be used not only for classification, but also for regression problems, pattern recognition, etc. The specific idea of the method is that a sample belongs to a class with the majority of the K nearest samples in the feature space, then the sample is also assigned to this class. The K-nearest neighbor algorithm is defined as, given a labeled training dataset, the unlabeled new sample is assigned to the class to which the majority of the K samples belong, given that the K samples in this training dataset are the nearest neighbors to the sample. This algorithm determines the class of the sample to be classified based on the class of the nearest K (one or several) samples, the basis for determining the attributed class is mainly based on the limited number of surrounding nearest neighbor samples rather than on the discriminative class domain. Therefore, it is a non-parametric classification algorithm for samples with overlapping or intersecting class domains.

The following methods are commonly used to determine the nearest neighbors for the K-nearest neighbor algorithm.

(1) Attribute nearest neighbor method. By finding the K nearest neighbors of a sample with unknown attributes (the attributes of these K neighbors are known), the attributes of these K neighbors are averaged and assigned to the sample.

(2) Distance nearest neighbor method. If the nearest K neighbors of an unclassified labeled sample are retrieved, and most of these K neighbors belong to a certain category, the unclassified labeled sample is assigned to this category.

(3) Weighted nearest neighbor method. In the labeled dataset, if the impact of the labeled sample is different from that of the sample to be labeled, the sample is labeled as the category to which most of the K samples in the dataset with the largest weight coefficient belong. For the weighted nearest neighbor method, there is the weighted nearest neighbor method alone (the distance is inversely proportional to the weight, and the larger the weight, the closer it is considered), and there is also the weighted nearest neighbor method that combines the weight and the distance, such as the distance multiplied by the inverse of the weight.

The calculation steps of the K-nearest neighbor algorithm are shown below.

(1) Obtain data from various databases and divide them into training data and validation data.

(2) Set the initial value of K in the K-nearest neighbor algorithm. There is no uniform method to determine the K value, and this paper adopts the K-fold cross-validation to determine the K value.

(3) By calculating the distance (generally using the Euclidean distance formula), the K samples in the training sample set X that are closest to the test sample x are found, and these K samples are denoted as $N_k(x)$. Let x_i^i denote the i-th feature attribute value of the i-th sample, $x_i(x_1^i, x_2^i, x_3^i, \dots, x_n^i) \in R^n$ (R^n is an n-dimensional space), then the Euclidean distance between these two samples x_i and x_j is shown in Eq. (1).

$$d(x_i, x_j) = (\sum_{l=1}^n (x_l^i - x_l^j)^2)^{\frac{1}{2}} \quad (1)$$

(4) Let y_i be the category label of x_i and $y_i \in Y = \{c_1, c_2, c_3, \dots, c_k\}$, where c_i is the i-th label category, and determine the category y of x in $N_k(x)$ according to the classification rules.

(5) Change the K value and go to step (3), and calculate the classification accuracy until the K value with higher classification accuracy or stop the cycle after reaching the specified number of cycles.

(6) Finally, the classification result of the K value with the smallest error rate is selected.

A flowchart of the KNN algorithm classification process is shown in the Fig. 2.

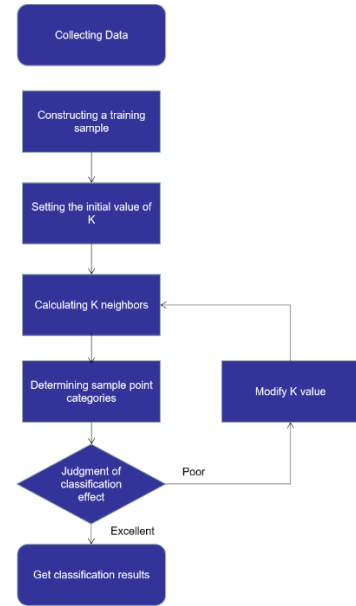


Fig. 2 Flowchart of KNN classification

The KNN algorithm is one of the simplest machine learning methods and has the following main advantages.

(1) The KNN algorithm is intuitively simple and easy to implement.

(2) The algorithm has a high tolerance for outliers and a high tolerance for noise because it selects K neighboring values to determine the classification, i.e., noise has little effect on the effect of the classification algorithm.

(3) No complex training process is required, and it is an "online" technique, where new sample data can be added directly to the sample data set without retraining.

(4) It supports multiple classifications with high accuracy, and can also solve the non-linear regression problem.

(5) The classification process of KNN algorithm directly utilizes the relationship between samples, thus reducing the

impact of improper selection of class features on the classification results and minimizing the error term in the classification process.

Of course, the algorithm also has some drawbacks.

(1) It is computationally intensive and slow. Because each sample to be classified needs to calculate its distance to all known samples in order to find the K neighbors of the sample. The common solution is to pre-process the sample data by first cropping the known sample points and removing the samples that are not useful for classification. This algorithm is suitable for automatic classification of class domains with relatively large sample data. The algorithm is not suitable for small sample sizes, because the training samples may be incomplete after cropping the useful data and affect the classification effect.

(2) The classification accuracy is highly dependent on the K-value. A randomly selected K-value may lead to misclassification or large errors, and it is usually necessary to select K-values several times to achieve the best classification results. As shown in Fig. 3, the data set has two classes (represented by triangles and circles). When K is taken as 3, the data points to be measured (represented by small five-pointed stars) should belong to the triangle number as shown in the figure, and if K is taken as 5, they will be classified as circles.

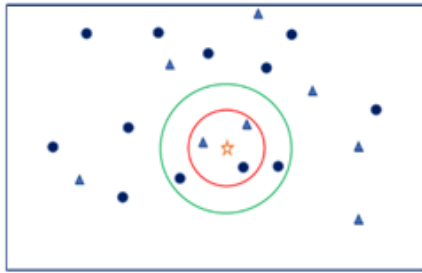


Fig. 3 Influence of K-value

(3) Sample imbalance is prone to classification misclassification. In the sample data set, if the sample data of one class is large and the sample data of other classes are small, it is possible that the majority of the K nearest neighboring samples of an input unknown sample to be tested is the class with large sample data, and the sample is classified as this class, but this class is not close to the target sample, while the number of samples of which classes are small is closer to the target sample, and leads to inaccurate classification.

C. Decision Tree Classifier

Decision trees are one of the powerful methods commonly used in various fields, such as machine learning, image processing and pattern recognition [11]. It is a tree structure in which each internal node represents a test on an attribute, each branch represents a test output, and each leaf node represents a category. Decision tree learning is a top-down recursive method, the basic idea of which is to construct a tree with the fastest decreasing entropy value as a measure of information entropy, with zero entropy value at the leaf nodes, at which point the instances in each leaf node belong to the same class. Most of the decision tree algorithms in use today center on optimizing some impurity metric in a branching approach. Impurity is calculated based on nodes, and each node in the tree will have an impurity, and the

impurity of a child node must be lower than that of the parent node. That is, the impurity of a leaf node must be the lowest in the same decision tree. It has two calculation methods to determine the impurity, Gini coefficient and information entropy, which are calculated as follows in Eq. (2) and (3).

$$Entropy(t) = -\sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t) \quad (2)$$

$$Gini(t) = 1 - \sum_{i=0}^{c-1} p(i|t)^2 \quad (3)$$

$p(i|t)$ represents the probability that node t is category i . The information entropy is more sensitive to impurity than the Gini coefficient. However, in practical use, information entropy and Gini coefficient have essentially the same effect. The calculation of information entropy is somewhat slower than the Gini coefficient because the calculation of the Gini coefficient does not involve logarithms. In addition, because the information entropy is more sensitive to impurity, the decision tree grows more "finely" when the information entropy is used as an indicator, so for high-dimensional data or data with a lot of noise, the information entropy is easily overfitted, and the Gini coefficient tends to work better in this case. Of course, this is not absolute.

The decision tree's structure is shown in the following Fig. 4 [12].

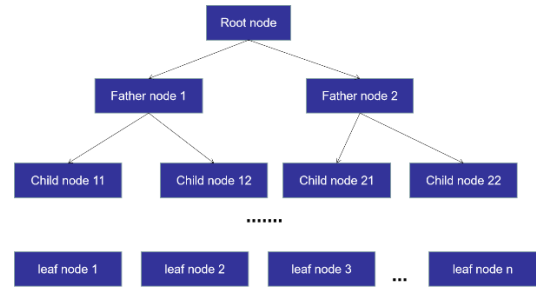


Fig. 4 Structure of decision tree

Some of the parameters of the decision tree classifier section are described.

(1) criterion: feature selection criteria, [entropy, gini]. Default 'gini', which is CART algorithm.

(2) splitter: feature division criteria, [best, random]. 'best' finds the best division point among all division points of the feature, 'random' finds the local best division point among some division points. Generally speaking, the default 'best' is suitable when the sample size is small, and if the sample data size is very large, then 'random' is recommended for decision tree construction.

(3) max_depth: the maximum depth of the decision tree, [int, None]. The default value is 'None'. If the model has more samples and more features, it is recommended to limit this maximum depth, the specific value depends on the data distribution.

(4) max_leaf_nodes: the maximum number of leaf nodes, [int, None]. By setting the maximum number of leaf nodes, it can prevent overfitting. Default value 'None', the maximum number of leaf nodes is not set by default. If the limit is added, the algorithm will build the decision tree that is optimal within the maximum number of leaf nodes. If there are not many features, this value can be disregarded.

D. Introduction of experimental indicators

1.F1-score is a statistical measure of the accuracy of a binary classification model. The F1 score can be considered as a weighted average of the precision rate and recall of the model, with a maximum value of 1 and a minimum value of 0. It is calculated as follows in Eq. (4).

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

Precision stands for model precision rate, Recall stands for model recall. The formula for both is as follows in Eq. (5) and Eq. (6).

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

TP stands for True Positive, meaning the number of positive samples judged to be positive; FP stands for False Positive, meaning the number of negative samples judged to be positive; FN stands for False Negative, meaning the number of positive samples judged to be negative.

2. Accuracy is the accuracy of image classification, calculated as follows in Eq. (7).

$$Accuracy = \frac{R}{S} \quad (7)$$

R stands for the number of correct verifications, S stands for the total number of verification sets.

3. The confusion matrix is to count the number of observations in the wrong and right categories in the classification model separately, and then display the results in a table. It has four basic indicators TP, FP, FN, and TN (TN stands for True Negative, meaning the number of samples judged to be negative and in fact negative). The four indicators are presented together in a table to form a confusion matrix, as shown in the Fig. 5.

Confusion matrix		Practice	
		Positive	Negative
Prediction	Positive	TP	FP
	Negative	FN	TN

Fig. 5 Confusion matrix

III. Experiment process

A. KNN image classification experiment

First, download the csv file of Fashion MNIST dataset from tensorflow, where each line contains one label category (ten categories are represented using 0-9 numbers respectively) and 784 pixel values. For KNN classification, the choice of K value is crucial, and this paper uses K-fold cross-validation to choose a K value of 4. K-fold cross-validation is to divide the original data into K groups, and make each subset of data into a validation set, and the remaining K-1 subset of data as the training set, so that K models will be obtained. The results of each of the K models are evaluated in the validation set, and the value of K with the smallest error is taken out and used for the initial parameters we set, thus achieving parameter optimization.

In this paper, 60,000 images are used as the training set and 10,000 images are used as the test set. The validity of the K-fold cross-validation method was verified by selecting k values from 1-10. The highest accuracy is achieved when k is

chosen to be 4. The results of KNN image classification experiments are shown in Table I and the confusion matrix and classification report are shown in Fig. 6 and Fig. 7.

Table I. Results of KNN classification

K	Accuracy	F1-score
1	0.8491	0.8503
2	0.8497	0.8503
3	0.8561	0.8563
4	0.8597	0.8599
5	0.8577	0.8576
6	0.8575	0.8575
7	0.8541	0.8540
8	0.8554	0.8553
9	0.8530	0.8528
10	0.8538	0.8535

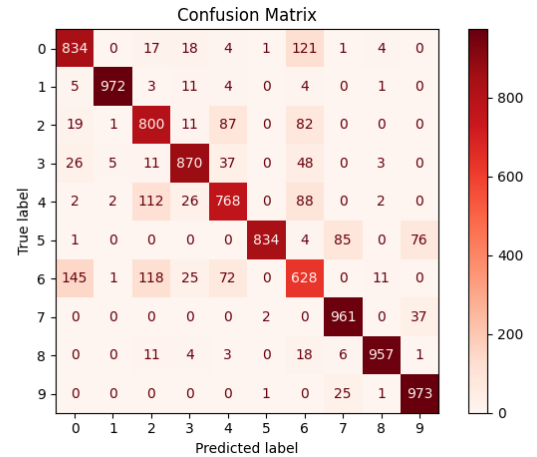


Fig. 6 Confusion Matrix from KNN classification

Classification Report:				
	precision	recall	f1-score	support
0	0.81	0.83	0.82	1000
1	0.99	0.97	0.98	1000
2	0.75	0.80	0.77	1000
3	0.90	0.87	0.89	1000
4	0.79	0.77	0.78	1000
5	1.00	0.83	0.91	1000
6	0.63	0.63	0.63	1000
7	0.89	0.96	0.92	1000
8	0.98	0.96	0.97	1000
9	0.90	0.97	0.93	1000
accuracy			0.86	10000
macro avg	0.86	0.86	0.86	10000
weighted avg	0.86	0.86	0.86	10000

Fig. 7 classification report for KNN

From the classification report it can be seen that the number of each category is 1000. Category 5 has the highest precision index; category 1 and category 9 have the highest recall index; category 1 has the highest F1-score. Overall, category 1 has the best recognition and KNN has good classification results.

B. Decision tree classifier experiment

Similarly, the decision tree classifier also uses the csv file of Fashion MNIST for image classification experiments. For the decision tree classifier parameter setting, the max depth parameter is set to 10 after several experiments due to the large data set, and the rest parameters are set to default values. All parameters are listed in the Table II.

Table II. Parameters of decision tree classifier

criterion='gini',	splitter='best',	max_depth=10,
min_samples_split=2,	min_samples_leaf=1,	
min_weight_fraction_leaf=0.0,	max_features=None,	
random_state=None,	max_leaf_nodes=None,	
min_impurity_decrease=0.0,	min_impurity_split=None,	
class_weight=None,	presort=False	

Then the image classification experiments were conducted, and the F1-score and Accuracy results are shown in Table III, and the confusion matrix and classification report are shown in Fig. 8 and Fig. 9.

Table III. Results of decision tree classifier

Accuracy	F1-score
0.8006	0.8013

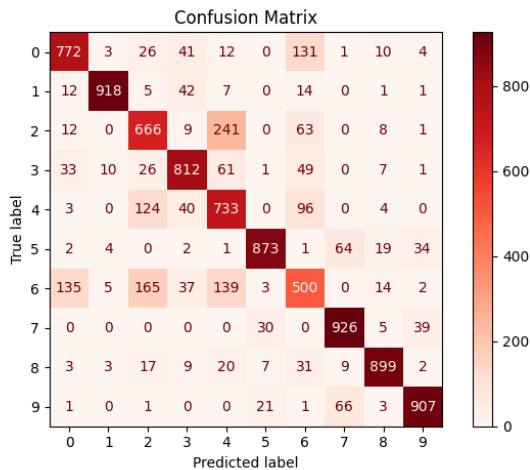


Fig. 8 Confusion Matrix from decision tree classifier

Classification Report:				
	precision	recall	f1-score	support
0	0.79	0.77	0.78	1000
1	0.97	0.92	0.94	1000
2	0.65	0.67	0.66	1000
3	0.82	0.81	0.82	1000
4	0.60	0.73	0.66	1000
5	0.93	0.87	0.90	1000
6	0.56	0.50	0.53	1000
7	0.87	0.93	0.90	1000
8	0.93	0.90	0.91	1000
9	0.92	0.91	0.91	1000
accuracy			0.80	10000
macro avg	0.80	0.80	0.80	10000
weighted avg	0.80	0.80	0.80	10000

Fig. 9 classification report for decision tree

From the classification report it's clear that category 1 has the highest precision index, recall index and F1-score. It can also identify 1 category very well. But the decision tree

classifier is slightly worse than KNN in terms of overall classification effect.

C. Results discussion

According to the experimental results derived from the above two classifiers, KNN and decision tree classifier, it can be seen that KNN is slightly better than decision tree algorithm for image recognition of Fashion MNIST dataset. However, it is undeniable that the decision tree algorithm outperforms the KNN algorithm in terms of correct recognition rate in some categories. Therefore, from the perspective of evaluation metrics, KNN algorithm can be well implemented for classification of Fashion MNIST dataset.

IV. CONCLUSION

In this paper, two classification algorithms, KNN and decision tree are used and image classification experiments are conducted based on the Fashion MNIST dataset. Firstly, a development history of image classification techniques and the Fashion MNIST dataset are introduced. Then the basic principles, general structure and some parameters of KNN and decision tree algorithms are analyzed in detail. The formulae and basic meanings of the three evaluation metrics (F1-score, Accuracy, and Confusion matrix) required for image classification experiments are also given. After that, experiments are conducted using KNN and decision tree for the dataset and the results are discussed. Finally, it is concluded that the KNN algorithm is slightly better than the decision tree algorithm for image classification, it's more suitable for Fashion MNIST classification.

REFERENCES

- [1] Abdulqader D M, Abdulazeez A M, Zeebaree D Q. Machine learning supervised algorithms of gene selection: A review[J]. Machine Learning, 2020, 62(03): 233-244.
- [2] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.
- [3] Xu Y, Li Z, Yang J, et al. A survey of dictionary learning algorithms for face recognition[J]. IEEE access, 2017, 5: 8502-8514.
- [4] Herath T, Rao H R. Encouraging information security behaviors in organizations: Role of penalties, pressures and perceived effectiveness[J]. Decision Support Systems, 2009, 47(2): 154-165.
- [5] You S, Xu C, Xu C, et al. Learning with single-teacher multi-student[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2018, 32(1).
- [6] Charbuty B, Abdulazeez A. Classification based on decision tree algorithm for machine learning[J]. Journal of Applied Science and Technology Trends, 2021, 2(01): 20-28.
- [7] Li B, Chen Y W, Chen Y Q. The nearest neighbor algorithm of local probability centers[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2008, 38(1): 141-154.
- [8] Patil D V, Bichkar R S. A hybrid evolutionary approach to construct optimal decision trees with large data sets[C]//2006 IEEE International Conference on Industrial Technology. IEEE, 2006: 429-433.

- [9] Deng L. The mnist database of handwritten digit images for machine learning research [best of the web][J]. IEEE signal processing magazine, 2012, 29(6): 141-142.
- [10] Xiao H, Rasul K, Vollgraf R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms[J]. arXiv preprint arXiv:1708.07747, 2017.
- [11] Stein G, Chen B, Wu A S, et al. Decision tree classifier for network intrusion detection with GA-based feature selection[C]//Proceedings of the 43rd annual Southeast regional conference-Volume 2. 2005: 136-141.
- [12] Fu P, Ma C, Li F, et al. Decision Tree classifier of coal & gangue Stack Areas[C]//2011 4th International Congress on Image and Signal Processing. IEEE, 2011, 3: 1619-1622.