Advanced Machine Learning CW1 Report

Hojin Lee, 20205232 [*1]

1. Introduction

This report contains experiments with different machine learning techniques for simple image classification task.

2. Decision Tree

2.1. Algorithm description and implementation details

This section presents the implementation details of decision trees and the results of the experiments from two different data sets. Decision trees classifier breaks up a complex decision into a union of several more straightforward decisions. There are mainly three steps for designing the trees, and below is the detailed implementation of such steps;

- The selection of a node splitting rule: we use the univariate splitting function; hence, each node splits according to the value of a single attribute. And we use information gain-based attributes to set the splitting rule. A random sampling of threshold and feature space are applied to reduce the training time as we have more than three thousand feature vectors from image data.
- Conditions to check if the node is the leaf: We have three rules to decide whether the node is a terminal leaf node or not; i) check the depth of node and terminate it if it reaches the pre-set maximum depth of the tree; ii) check if the number of samples is less than two; iii) check if the number of labels in the samples is only one.
- Selection of class label in the terminal node: the class of the terminal node is assigned to have the highest probabilities. Further, probabilities are estimated by the ratio of samples in the terminal node.

2.2. Model parameters

To test the parameter performance, we used cross validation. The selected parameters are; i) depth of Tree; ii) number of sampled features; iii) number of sampled threshold.

2.3. Algorithm performance analysis

The CIFAR-10 dataset has 50000 color images where each image consists of 3072 feature vectors. We first sampled 5000 data to train. The accuracy on train and test data with different parameter settings are plotted in Fig. 1, and 2, respectively.

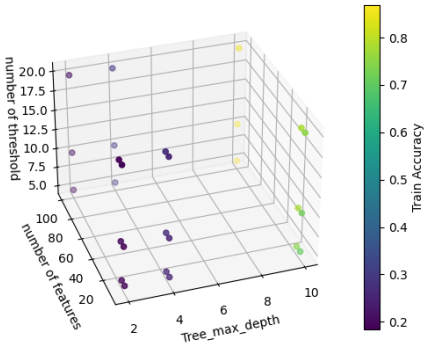

Figure 1. Train data accuracy of decision trees with 5000 samples (CIFAR-10)

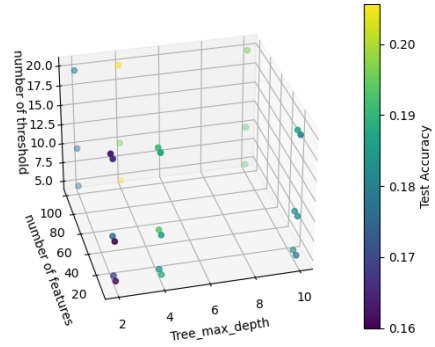

Figure 2. Test data accuracy of decision trees with 5000 samples (CIFAR-10)

The best accuracy is achieved for both training and testing results when all the parameters values are high (e.g., the dense tree is better than the shallow tree, more features and thresholds are better). We could see the training accuracy approaching one as we increase the values of the parameters, which indicates the successful model training on the given data. However, the test accuracy is poor. This is expected

as we have only utilized a maximum of 20 features for the training sample, which originally had about 3000 features per image. Also, we have only used 5000 sampled data. Clearly, the model is not generalized enough on the test data.

The Fig. 3, and 4 show the results of the trained model with the entire data set. Even though the accuracy increased when we had higher values of all parameters, which is a similar trend from the previous results, the training is failed to achieve admissible accuracy. We have suspected that it is because the number of features used in the training process is well below the dimension of feature space of the original data. Yet, the best test result of the trained model is better than the previously trained model with less number of samples. We could have used all the features and thresholds to train the better model. However, due to the limited resources and time, we did not proceed further experiments with a wider range of parameter settings.

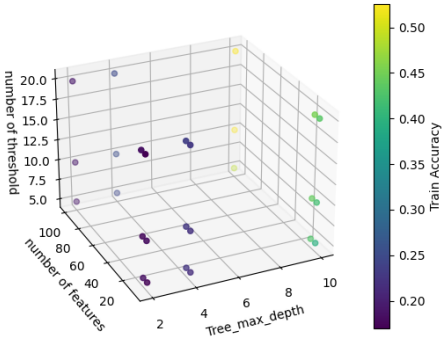

Figure 3. Train data accuracy of decision trees with 50000 samples (CIFAR-10)

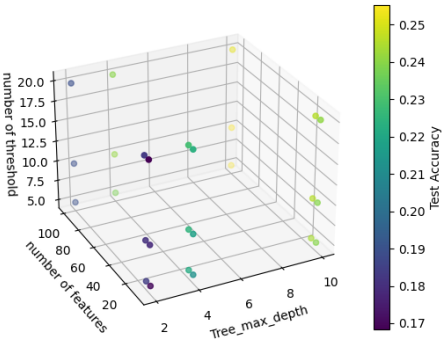

Figure 4. Test data accuracy of decision trees with 50000 samples (CIFAR-10)

By applying the same strategy, we also trained the model on CIFAR-100 dataset. And we have recorded the best results among different parameter set in Table 1.

Table 1. Classification accuracies of Decision trees with various data sets. Each scenario runs the hyperparameter tuning and 5-fold cross validation, and picks the best results.

DATA SET	NUMBER OF SAMPLE	ACCURACY (TRAIN)	ACCURACY (TEST)
CIFAR-10	5000	0.852	0.205
CIFAR-10	50000	0.526	0.252
CIFAR-100	5000	0.866	0.030

3. Random Forest

3.1. Algorithm description and implementation details

This algorithm takes benefits from having multiple decision trees in the model. The prediction result is the most votes from multiple decision trees. Each individual decision tree is trained by differently sampled data as well as parameter settings. It is known that having such multiple models can improve performance in general because different models protect each other from their individual errors. This is helpful, especially since Decision trees are sensitive to training data set. We have applied the same strategy in Section 2.1 for individual decision trees to train the model.

3.2. Model parameters

To test the parameter performance, we used cross-validation. Since we did not see vast improvements from the increased number of thresholds in Decision trees models, we fixed the number of sampled thresholds as 10. And the rest of the selected parameters are as follows; i) the number of trees; ii) the depth of Tree; ii) the number of sampled features.

3.3. Algorithm performance analysis

We present the results on the CIFAR-10 dateset in Fig. 5, and 6. As it is expected, more trees give better results in testing data. It is because multiple uncorrelated trees help each other improve the performance in general. Similar to the decision trees, we could see the improvement with more depth and more features for computing the information gains. We also trained the model on different dataset, and the best results among different parameter sets are given in Table 2.

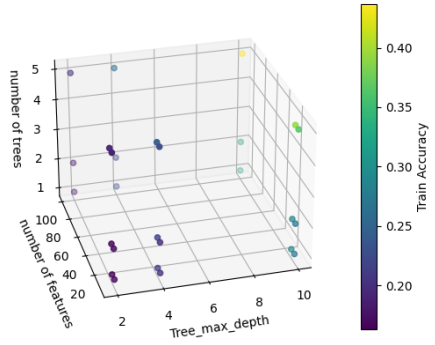

Figure 5. Train data accuracy of Random Forest (CIFAR-10)

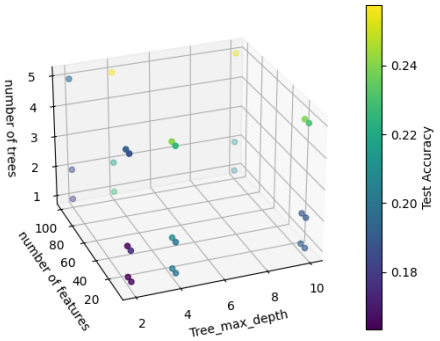

Figure 6. Test data accuracy of Random Forest (CIFAR-10)

Table 2. Classification accuracies of Random Forest with various data sets. Each scenario runs the hyperparameter tuning and 5-fold cross validation, and picks the best results.

DATA SET	NUMBER OF SAMPLE	ACCURACY (TRAIN)	ACCURACY (TEST)
CIFAR-10	5000	0.435	0.200
CIFAR-10	50000	0.421	0.296

4. Support Vector Machine

4.1. Algorithm description and implementation details

Support Vector Machine (SVM) has great advantages by having a maximum margin for differentiating different classes while separating samples using hyperplanes. It is a well-known fact that this method is robust to outliers by ignoring some features while maximizing the margin. The Kernel is used to have non-linear hyperplanes, and we use two different kernels; i)linear ii)radial basis function (RBF). To avoid an over-fitting problem, having a simple kernel while separating high-dimensional data is important. About the implementation, we import the corresponding algorithm from **sklearn** library.

4.2. Model parameters

To test the performance of the trained model with different parameters, we used cross-validation like other training procedures. We have trained SVM with two different kernels, and each has a different set of parameters. First, the linear kernel is adopted with parameter 'C', which represents the regularization. Lowering the value of 'C' will be a more smooth function, hence generalized for testing performance. Secondly, 'Radial Basis function(RBF)' kernel is used with two parameters (e.g., C, and Γ). It is worth noting that 'C' represents the regularization and Γ defines how much influence a single training sample has. For example, the larger gamma is, the closer other examples must be to be affected. Therefore, the higher the Γ is, it is more subjective to the outliers in sample.

4.3. Algorithm performance analysis

Firstly, the results on CIFAR-10 dataset with RBF as kernel is in Fig. 7. Different values of 'C', which implies the regularization, affects the accuracy in training session. Lower value of 'C' makes too smooth function for the trained model so it cannot capture the true dynamics of samples, hence it has low accuracy even in training data. However, having less regularization makes model over fitted to the training data which can be shown by that the accuracy on training data reaches one while accuracy on test data is poor.

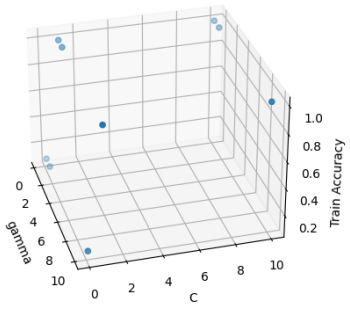

Figure 7. Train data accuracy of Random Forest with 5000 samples (CIFAR-10)

Table 3. Classification accuracies of Support vector machines with various regularization 'C'.

KERNEL	C	ACCURACY (TRAIN)	ACCURACY (TEST)
RBF	0.1	0.471	0.448
RBF	1	0.982	0.555
LINEAR	0.1	1.0	0.299
LINEAR	1	1.0	0.300

Table 4. Classification accuracies of Support vector machines with various data sets. Each scenario runs the hyperparameter tuning and 5-fold cross validation, and picks the best results.

KERNEL	NUMBER OF SAMPLE	ACCURACY (TRAIN)	ACCURACY (TEST)
RBF	5000	1.0	0.100
RBF	50000	0.526	0.2 52

5. K-nearest neighbor

5.1. Algorithm description and implementation details

This section presents the training results using k-nearest neighboring (KNN) classifiers, which classify data points based on what are most close to it based on the distance between features of samples. To train the model, the number of groups needs to be given at the beginning of the training session (e.g., 10 groups in CIFAR-10 data, as there are 10 different labels in the sample). The major bottleneck of this method is that it has to use training data to make new inferences as it has to calculate the distance from individual samples again. Therefore the runtime depends on the number of training samples. To reduce the runtime, we also tried to apply the k-mean clustering algorithm for classification problems. The main difference is that k-mean clustering has center clustering points representing the average of the features within the same group of the samples. Therefore, only the number of groups' feature vectors is used during the runtime to calculate the distances to the inferring data. Then the label is given as the one which has the closest distance to that sample.

5.2. Model parameters

There are two different parameters for KNN; i) the number of iteration for clustering the samples; ii) number of nearest neighbor (e.g., we denote it as 'kparam'). Each new sample calculates the distance from 'kparam' number of closest samples and decide its labels depends on the most common labels within the selected samples.

5.3. Algorithm performance analysis

Table 5. Classification accuracies of **K-nearest neighbor** with CIFAR-10 data sets.

NUMBER OF SAMPLE	KPARAM	ACCURACY (TRAIN)	ACCURACY (TEST)
5000	10	0.169	0.157
5000	100	0.173	0.161

Table 6. Classification accuracies of **K-mean clustering** with CIFAR-10 data sets.

NUMBER OF SAMPLE	ACCURACY (TRAIN)	ACCURACY (TEST)
5000	0.172	0.102

6. Convolutional Neural Network

This section, we extend our study with neural network model for image classification. The structure of the trained model is based on the architecture of VGG16 network (Simonyan & Zisserman, 2014). The model consists of 13 convlutional, 5 maxpooling, and final dense layer to get probabilities for different labels. The sourcecode is based on the implementation from (Geifmany, 2021). The accuracy of the trained model reaches 0.9362 for validation set of CIFAR-10, and 0.7049 for CIFAR-100. It is clear that this state of art neural network surpasses the others from previous sections.

References

Geifmany. Vgg16cifar. https://github.com/geifmany/cifar-vgg.git, 2021.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.