INFS7203 Data-oriented Project Proposal

Yating Zhang
School of Information Technology and Electrical Engineering
The University of Queensland, Qld., 4072, Australia

Abstract

This project aims to develop a classifier based on the given dataset "train.csv", which could accurately classify unseen data points into one of the ten classes.

Appropriate pre-processing techniques need to be applied and will be determined by using cross-validation. The procedure of the four data classification techniques (decision tree, random forest, k-nearest neighbour and naïve bayes) will be described. The ensemble of the classification results from different classifiers will also be considered. The most appropriate metric for measuring the performance of the dataset is the F1-score. The last section is the timeline which ensures the project could be done smoothly on time.

1 Pre-processing Techniques

Based on the initial look at the dataset, the pre-processing techniques are necessary to be applied before classification.

Cross-validation can be applied to evaluate the performances of different methods when there is more than one method for different pre-processing techniques. For a training dataset D, we randomly divided it into K folders with almost equal number of instances in each folder.[1] Every time D_i is used as validation data and all others (D_1, ... , D_{i-1}, D_{i+1}, ... , D_k) are used as training data. Each validation set will generate a validation score. The average of all the K scores gives the performance of a method.

Imputation is required as there are missing feature values throughout every feature in the dataset. For example, the model could benefit from imputation because feature Num_Col_0 has 205 missing feature values, and the missing rate is 0.094. For different value types[1], the missing numerical values could be imputed by the average of the feature's values, the missing nominal values could be imputed by the most common feature value. There are two imputation methods, imputation by the feature's all values and imputation by class-specific values. Cross-validation will be used to determine which imputation method suits this project.

Normalization is required as features in this dataset have different degrees of magnitude [1]. For example, the feature Num_Col_0 has an average value of -1.9, but the feature Num_Col_22 has an average value of 4863.3. The magnitudes may impact the classification, so normalization of these feature values is required. There are two methods for data normalisation, max-min normalization and standardization. The max-min normalization is helpful when data does not follow a Gaussian distribution. The standardization is helpful when data follows a Gaussian distribution. Cross-validation will be used to determine which normalisation method is more suitable.

There are many outlier detection methods, for example, density-based technique, model-based technique, distance-based technique, cluster-based technique, and isolation-based technique.[2]

As it is hard to evaluate the performance of the outlier detection at the initial stage, I will do it after I have confirmed all the other hyperparameters. If the classifier's performance gets better with outliers removed, I will keep the outlier technique and tune the parameter, otherwise, I will not remove any outliers.

In conclusion, these pre-processing techniques, imputation, normalization, and outlier detection are all pre-processing techniques that could be applied to this dataset. The cross-validation could be used to determine the appropriate methods by evaluating the preferences. It is crucial as all the classification techniques after that are based on the pre-processed data.

2 Decision Tree

Applying a decision tree to the dataset is a recursive procedure. Firstly, generate two sets, the training set D and feature A. Secondly, select the optimal splitting feature a* from A, for all possible values of a*, generate a branch and a subset, and keep splitting until stop criteria have reached [3]. The optimal splitting feature a* is determined by the increasing purity and it could be measured by information gain, gain ratio and Gini index.

The entropy describes the level of "disorder", and the information gain is the entropy reduction after a splitting. Selecting the optimal splitting feature by the one having the largest information gain. The equations for information gain:

$$Gain(D,a) = Ent(D) - \sum_{j-1}^{V} \frac{|D_{vj}|}{|D|\ Ent(D_{vj})}$$

$$\mathrm{Ent(D)} = -\sum_{\mathrm{i=1}}^{\mathrm{m}} \mathrm{p_i} \log_2 \mathrm{p_i}$$

A shortcoming of information gain is that it could result in too many branches, and the gain ratio can help reduce the chance of having too many branches. The equations for gain ratio:

$$\text{Intrinsic value: } \mathrm{IV(D,a)} = -\sum_{j=1}^{V} \frac{|D_{vj}|}{|D|} log_2 \frac{|D_{vj}|}{|D|}$$

$$Gain\_ratio(D,a) = \frac{Gain(D,a)}{IV(D,a)}$$

The gini index gives the likelihood of two samples randomly selected from the dataset belonging to different classes.

The equations for the gini index:

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2$$

$$Gini\_index(D,a) = \sum_{j=1}^{V} \frac{|D_{vj}|}{|D|} Gini(D_{vj})$$

For continuous-valued features, the features can be split using bi-partition. The main procedure is to sort all values and the candidate split values, then calculate the purity for each value based on purity measures.

Pruning off some of the branches in the decision tree could reduce the risk of overfitting. Pre-pruning and post-pruning are two different pruning approaches. The Pre-pruning suits the dataset with which we have domain knowledge, otherwise it may lead to potential underfitting and information loss. On the other hand, post-pruning tends to yield higher accuracy but comes at the cost of computational resources. The cross-validation needed to be applied during the process of pruning.

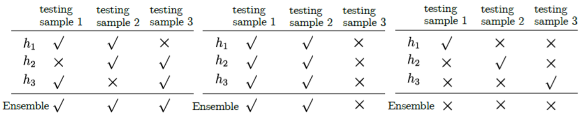
3 Random Forest

Random forest is an ensemble method that construct multiple decision trees by applying bootstrap sampling [4] to obtain the data subsets, and randomly sampling k features among all the features, then combining the output of all the decision trees to achieve a better prediction.

Bootstrap sampling is used for generating different subsets. Given a training data set containing m number of training examples, a sample of m training examples will be generated by sampling with replacement. Some original examples appear more than once, while some original examples are not present in the sample.[4]

During the construction of a component decision tree, at each step of split selection, Random Forest first randomly selects a subset of features and then carries out the conventional split selection procedure within the selected feature subset.[4]

The performance of the random forest can also be evaluated using cross-validation.



	testing sample 1	testing sample 2	testing sample 3
h_1	√	√	×
h_2	×	√	√
h_3	√	×	√
Ensemble	√	√	√

(a) Improved performance.

	testing sample 1	testing sample 2	testing sample 3
h_1	√	√	×
h_2	√	√	×
h_3	√	√	×
Ensemble	√	√	×

(b) Unchanged performance.

	testing sample 1	testing sample 2	testing sample 3
h_1	√	×	×
h_2	×	√	×
h_3	×	×	√
Ensemble	×	×	×

(c) Degenerated performance.

Fig 1. An Example of Ensemble Learning [3]

4 K-NEAREST NEIGHBOURS

K-nearest neighbours is a lazy learning approach, which makes predictions using historical records and applies the same label with the most similar record.

The K-nearest neighbours classifier needs three inputs, the training data, the distance matrix, and the hyperparameter K.

Before implementing the K-NN classifier, I will do data normalization. The distance matric I will use is the Euclidean distance. The K value is important because it impacts the sensitivity to noise data points. I will use cross-validation to tune the hyperparameter K.

The main procedure of K-NN is to compute the test data's distance to all the training samples, identify the nearest K-nearest neighbours, and then use the labels of the nearest neighbours to determine the label of the test data by majority vote.[5]

The most important hyperparameter for KNN is the number of neighbours, which is usually chosen from odd numbers from 1 to 21 [6]. I will start from the middle value of the range, choose K from 7 or 9 as the initial value of K and tune it using cross-validation.

5 Naïve Bayes

The naïve Bayes method follows the naïve attribute conditional independence, it assumes that all features are conditional independent from each other.

To predict the label for data x, calculate the $p(y|x)$ using:

$$p(y|x) \sim p(y) p((x_1|y)) p((x_2|y)) \cdots p((x_d|y))$$

The label is y which has the maximum value of $p(y|x)$.

If any zero probability exists, the Laplacian correction needs to be used. There are nominal features in the given dataset "train.csv", which means that the zero probability may exist, so I will use the Laplacian correction while calculating the probabilities. The equation for the Laplacian correction:

$$\mathrm{p}(y) = \frac{\text{the number of training samples with label } y + 1}{\text{the number of all training samples} + \text{the number of labels}}$$

$$\mathrm{p}(x_i|y) = \frac{\text{the number of training samples with feature } x_i \text{ and label } y + 1}{\text{the number of training samples with label } y + \text{the number of feature valus of the } i\text{th feature}}$$

Fig 2. The Laplacian Correction [5]

6 Ensemble of The Classification Results

Ensemble learning is to combine the output of multiple individual classifiers. The objective of the ensemble is to achieve a better prediction performance. There are some aspects that can impact the performance of an ensemble, for example, the ensemble diversity and the performance of each individual model.

Ensemble diversity, that is, the difference among the individual learners, is a fundamental issue in ensemble methods.[4] For example, the decision tree and random tree are tree-based methods, the K-NN is an instance-based method, and the Naïve Bayes is the probability-based method. Different methods have different advantages and disadvantages, ensemble diversity in theory could make higher prediction performance.

The performance of each individual model could also impact the final performance. If one of the models has only 30% accuracy, not only it does not help the performance of the ensemble results but pulls down the performance of the ensemble results.

In this project, I will start with a combination of two different models, then add more if the computational resources allow. Those individual models that have an accuracy higher than 67% could be considered. Finally, I will use cross-validation to assess their prediction performance.

7 Evaluation

The initial observation of the dataset (Figure 3) shows that the given dataset is an imbalanced dataset. Label-5 only has a 2% proportion in the dataset, which is very low as compared to the other labels.

Accuracy disregards class balance and the cost of different errors, [7] so accuracy does not suit this given subset.

In this case, the F1-score is the most appropriate evaluation metric to deal with the given imbalanced data.

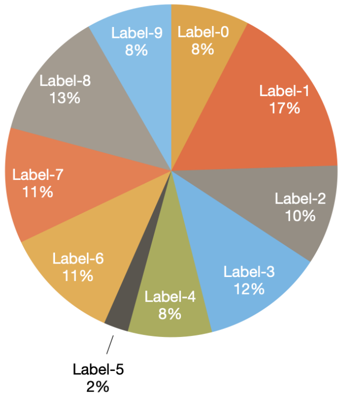


Fig 3. The Proportion of Each Label

8 Timeline

- First Milestone
  - Week 9 (18/9 to 24/9)
  - Pre-processing
- Second Milestone
  - Week 10-11 (25/9 to 8/10)
  - Classification and ensemble
- Third Milestone
  - Week 12 (9/10 to 15/10)
  - Evaluation and adjustment
- Forth Milestone:
  - Week 13 (16/10 to 22/10)
  - Test and Report

References

[1] Dr Miao Xu, "Lecture 2: Introduction to Classification," *University of Queensland, Data Mining (INFS4203/7203),* 2023.

[2] Dr Miao Xu, "Lecture 7: Anomaly Detection," *University of Queensland, Data Mining (INFS4203/7203),* 2023.

[3] Dr Miao Xu, "Lecture 3: Decision Tree and Random Forest," *University of Queensland, Data Mining (INFS4203/7203),* 2023.

[4] Zhi-Hua Zhou, *Ensemble methods : foundations and algorithms*. 2012.

[5] Dr Miao Xu, "Lecture 4: k-Nearest Neighbors and Naïve Bayes," *University of Queensland, Data Mining (INFS4203/7203),* 2023.

[6] Jason Brownlee. "Tune Hyperparameters for Classification Machine Learning Algorithms." https://machinelearningmastery.com/hyperparameters-for-classification-machine-learning-algorithms/ (accessed.

[7] Evidently Ai Team. "Accuracy, precision, and recall in multi-class classification." https://www.evidentlyai.com/classification-metrics/multi-class-metrics (accessed.