Prediction of Music Origin

Musha Wen

Research School of Computer Science, Australian National University
u5962473@anu.edu.au

Abstract. In the research field of machine learning, existing researches on music mainly rely on subjective judgement made by human. We introduce a more objective approach to solve the problem in this field. We focus on how to predict the origin when given a piece of music. We firstly use neural network to implement our approach. Then we improve our outcome that we achieved from the neural network model by applying the theories of heuristic pattern reduction and evolutionary algorithm separately. We train and test our model with the Geographical Origin of Music Data Set. By using the model we established, we are able to predict the origin of a piece of music and the performance of this model can be improved by separately utilizing heuristic pattern reduction and evolutionary algorithm as well. We also compare the result of our model with the work of other research paper, which uses machine learning approaches to solve this problem, and find that our model cannot provide such a good result as machine learning algorithm but it can be more general in practice.

Keywords: geographical ethnomusicology · neural network · heuristic pattern reduction · evolutionary algorithm.

1 Introduction

In this chapter, we will firstly introduce the problem we addressed in this report. Motivation and some necessary background, which may help the readers understand our report, will be covered in section 1.2. We will offer the information about the dataset we used to solve the problem in section 1.3. At the end of this chapter, a brief outline of this report will be listed.

1.1 Problem Statement

In this report, we focus on a problem of predicting the origin of a piece of music. This problem aims to make the machine predict the origin of a piece of music by given some features of that music. Similar topic has been discussed. In [8], researchers have already tried to address this problem by using some machine learning algorithms, such as KNN and Random Forest Regression. We will use a completely different approach, which is based on neural network theories, in this report. We will be given a dataset, in which each sample consists of features and the real origin of this music. We will use 80 percent of the dataset as training set. After training the neural network, we will test it by the rest of the dataset and require the neural network to predict the origin of each test sample. The real origin of each music will be used for evaluating the predictions we get. We will use two techniques to improve the performance of our neural network: one is heuristic pattern reduction, the other is evolutionary algorithm. Comparison and analysis of the results we get from different methodologies will be covered in this report as well.

1.2 Motivation & Background

Music has become one of the most popular topic in the research area of machine learning. Geographical ethnomusicology, which studies the distribution of music, is a branch of machine learning research in music. However, traditional research approaches of this topic, such as [8], based on the subjective judgement of humans to evaluate the results. This kind of methods rely on the feelings of human and are not objective enough. Therefore, we hope to find a more objective characteristic to distinguish the origins of two different pieces of music. Geographical position of the origin is a good choice.

As for the research work of neural network, 70% of applications of neural network use back-propagation training algorithms, which has a weakness in training speed according to [2]. Thus, researchers begin to think about how to improve the performance of neural networks naturally. Many studies have been done on this topic, such as [4] and [5]. Gedeon and Harris pointed out in [3] that reduce the size of training set may be a good solution. In this project, we will talk more about this kind of solution and apply the theories on our own neural network to see if

the performance will be improved. To be more concrete, we will use the theory of [2] to make improvement, which is similar to the conclusion of Gedeon and Harris.
Another technique that can be applied on the neural network model so as to improve the training performance of the model is feature selection. According to [6], too many input features can lead to the increase of unnecessary connections in the neural network, which may slow down the speed of training. Evolutionary algorithm [1] can be utilized to decide which feature is appropriate for the training of neural network.

1.3 Dataset

We will run and test our approaches based on the Geographical Origin of Music Data Set [8], which is created by Fang Zhou, in this report. This dataset contains 1059 audio tracks from 33 different countries or areas. The creator has already extracted the features for each audio track with the help of MARSYAS [7]. Each audio track is represented by 70 numbers while the first 68 are features with a mean of 0 and the last two are the geographical position, which is represented by latitude and longitude value, of the origin of this track. No weighting or pre-filtering process operation has been applied on the dataset.

1.4 Report Outline

In order to demonstrate the problem clearly, the following part of this report will be divided into three chapters. In chapter 2, we will introduce our own approach to address the problem, together with the improved solutions under the guide of [2] and [1]. Analysis and comparison of the results we get from these two methodologies separately will be covered in chapter 3. These three results will also be compared with the existing results gained from the dataset in this chapter. In the last chapter, we will give a conclusion of this report and have a brief discussion on future work.

2 Method

This chapter mainly introduces three approaches that can be used to address this geographical ethnomusicology problem. For convenience, we need to do some data preprocessing task on the dataset and the relevant content is talked in section 2.1. In section 2.2, we will present a solution on the basis of a naive neural network. An improved solution with the help of [2] will be discussed in section 2.3. Section 2.4 briefly summarizes the content of this chapter.

2.1 Data Preprocessing

Considering the discontinuity of longitude which makes two geographical adjacent areas have totally different longitude, while one is $+180°$ and the other is $-180°$, we need to do some preprocessing operation on the data. We find that if we convert the original spherical coordinates (i.e., latitude and longitude) to Cartesian coordinates, we won't need to be worried about the issue above. Therefore, we use the equation below to do this task:

$$\begin{cases} x = (R\cos\varphi)\sin\lambda \\ y = (R\cos\varphi)\cos\lambda \\ z = R\sin\varphi \\ R = 6373 \end{cases} \tag{1}$$

In this equation, φ represents for the latitude while the longitude is denoted by λ. x, y, z are the corresponding values in ECEF coordinates. The radius of the earth is represented by R.

2.2 Neural Network Solution

In order to address the problem of predicting the origin of a piece of music, we employ a 4-layer feed-forward neural network. We split the dataset into two parts: the training set has 847 patterns, which contain80% of the information given by the dataset. The left patterns form a validation test set.
The network has 68 input units and 3 output units. The first hidden layer has 1024 units. The second hidden layer has 128 units and the last hidden layer has 16 units. We use sigmoid function as the active function. We use stochastic gradient descent to propagate error backwards. The loss function is calculated by the mean square error function, which is same as it in the research methods proposed by [8]. We trained this network 50 epochs. The result we get from this network will be discussed in next chapter.

2.3 Improved Solution with Heuristic Pattern Reduction

According to the theories and experiments in [2], heuristic pattern reduction may improve the performance of the neural network. It is demonstrated in [2] that if we reduce the size of training patterns and use the same validation set for testing, the neural network may be able to gain the similar result as the result gained under the network with full-size training patterns. Meanwhile, the time consumption on training may be reduced.
Therefore, on the basis of the naive neural network we've already built, we keep the validation test set same as the previous solution while we train the neural network with 80%, 75%, 67%, 50%, 33%, 25% size of the full-size training set separately. At the same time, we increase the original learning rate correspondingly to see the results and performance. The new learning rate for each network is calculated by the equation below:

$$LR_{new} = \frac{LR_{old}}{size} \tag{2}$$

In this equation, LR_{new} and LR_{old} represents for the new learning rate and the original learning rate separately. The percentage of the new training set in the full-size training set is denoted by $size$.
These new networks are also trained 50 epochs. The results we get from these networks will be discussed in next chapter.

2.4 Improved Solution with Evolutionary Algorithm

Since we want to make use of the outcome of evolutionary algorithm, we decide to regard the result of this algorithm as a mask. By applying this mask on the input feature of our neural network model, we can filter unnecessary features and only use the features that we picked out from the original input feature set so that training performance of our network model will be improved.
According to the previous considerations, we suppose that in the population of each generation, there are 20 individuals, each of which is a mask. These masks are used for filtering unnecessary features of the dataset. Fitness of each individual is calculated by the error of 20 different neural networks. Apart from the input layer neurons, these 20 neural networks are same as the network we introduced in section 2.2. We execute the evolutionary algorithm 10 iterations. In each iteration, there are four small process: crossover, mutation, calculation of fitness and selection. These four process happen in the order we introduced them in every iteration. Before we start to iterate, we initialize our original population first by an initialization process.
Initialization process aims to initialize our population before we start other operations. As we have previously mentioned, these individuals are masks to filter unnecessary features. Therefore, we start with randomly generating 20 strings. Each string is 68-bits long and represented by binary representations, which means that each bit of the strings is either 0 or 1. In a string, if the bit is 1, then the feature of the input features that has the same label as this bit will be kept, otherwise, the feature will be discarded and the corresponding data in both training set and testing set will also be deleted. After this step, we start our evolutionary process.
Calculation of fitness process calculates the fitness of every individual. We use the individuals as masks to filter unnecessary input features together with corresponding data from the training set and testing set. After this step, we get 20 different versions of input feature as well as different training sets and testing sets which are subsets of the original training set and testing set. We train these 20 different neural networks 25 epochs and calculate the error in the same way as we do in section 2.2. These errors are the values of the fitness of different individuals.
Crossover process aims to generate offspring for the individuals in current iteration. We use sexual one-point crossover operator to achieve our goal. We rank our individuals according to their fitness from low to high since the lowest fitness means the lowest error that the neural network generated when having a mask, which is defined by an individual, applied on the input features and dataset. We pick two individuals which are adjacent in ranking and use them as parents for crossover operator. By this operation, we can have 10 pairs of parents and generate 20 new individuals, each of which is the offspring of original individuals.
Mutation process offers the new individuals a possibility that they can mutate, which is conform to the mutation process in biological area. We set that each bit of the new individuals has a probability of 0.1 to change its value (i.e., changes from 1 to 0 or changes from 0 to 1).
Selection process is a key step to decide which individual can survive in next generation. We use non-deterministic linear sampling ranked-based selection operator. Before we start to select, we calculate the fitness of new individuals first. The fitness is calculated in the same way as we do for the original individuals. Then we rank these 40 individuals according to their fitness from low to high. Considering the tradeoff between diversity and fitness, we decide to keep the top ten individuals in the ranking first and do the selection operation on the rest 30 individuals. We repeatedly select 10 individuals from the last 30 individuals. These 10 individuals we selected together with the top

10 individuals we kept form the population of next generation. After this step, the new iteration starts with the calculation of fitness process.
With this algorithm, we picked out the most necessary features from the original input feature set. We trained and tested our neural network model with these features. The results we achieved will be discussed in next chapter.

2.5 Summary

In this chapter, we introduce three approaches to solve the problem of predicting the origin of a piece of music. We use a simple neural network to solve the problem. On the basis of the first solution, we use heuristic pattern reduction technique to improve the performance of the neural network. With the aid of evolutionary algorithm, we do a feature selection on the input features of our neural network model with the same purpose as the second solution. Comparison of the results of these three methodologies will be discussed in next chapter.

3 Results and Discussion

This chapter shows the results we get in the three approaches we introduced in previous chapter. Section 3.1 provides an overview of our evaluation methodology, including the calculation of loss function of our neural network model. In section 3.2, we will discuss and compare the results we get in our own approach as well as the improved solution. In section 3.3, we will compare our results with the conclusion of another research paper, which is written by the creator of the Geographical Origin of Music Data Set. A brief summary of this chapter will be given in section 3.4.

3.1 Overview

For comparing our outcome with that of other work based on this dataset conveniently, we use the same loss function as [8].
We calculate the great circle distance from the true position to our predicted position. We evaluate the performance of our neural network by mean square error of the great distances. Therefore, we need to convert our output from the Cartesian coordinates to spatial coordinates by the equation given below:

$$\begin{cases} \varphi^p = \arctan 2(z, \sqrt{x^2 + y^2}) \\ \lambda^p = \arctan 2(y, x) \end{cases} \tag{3}$$

In this equation, φ and λ represent for latitude and longitude separately while x, y, z are the output of our neural network.
The great circle distance can be calculated by the equation below:

$$\begin{cases} d(P^p, P^t) = 2R \times \arctan 2(\sqrt{a}, \sqrt{1-a}) \\ a = \sin^2(\frac{\varphi^p - \varphi^t}{2}) + \cos\varphi^p \cos\varphi^t \sin^2(\frac{\lambda^p - \lambda^t}{2}) \\ R = 6373 \end{cases} \tag{4}$$

In this equation, the great circle distance is defined as $d(P^p, P^t)$, in which P^p is the predicted position and P^t is the true position. φ^p and λ^p are the predicted latitude and longitude. φ^t and λ^t represent the true latitude and longitude. R is the value of earth radius.

3.2 Experimental Results

With the equations above, we compare the loss of different neural network trained with different training set that has different size. Figure 1 shows the result of comparison.

From the diagram, we can find that when the new training set has 80%, 75% and 66.67% training patterns of the full-size training set, the result is similar as the result we get from the original neural network. Nevertheless, when the size of training set keeps decrease, the result becomes worse and worse. Therefore, with the help of heuristic pattern reduction, the performance of neural network get improved.

Here comes another diagram (Figure 2) which shows us there is a relation between resample ration and training time.

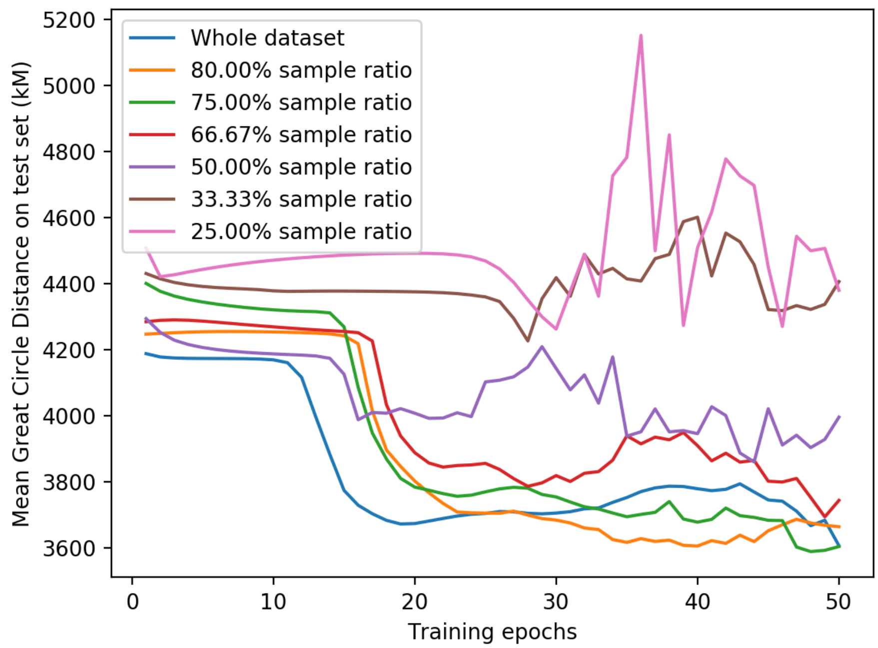

Fig. 1. Loss of Different Neural Network Trained with Training Set in Different Size

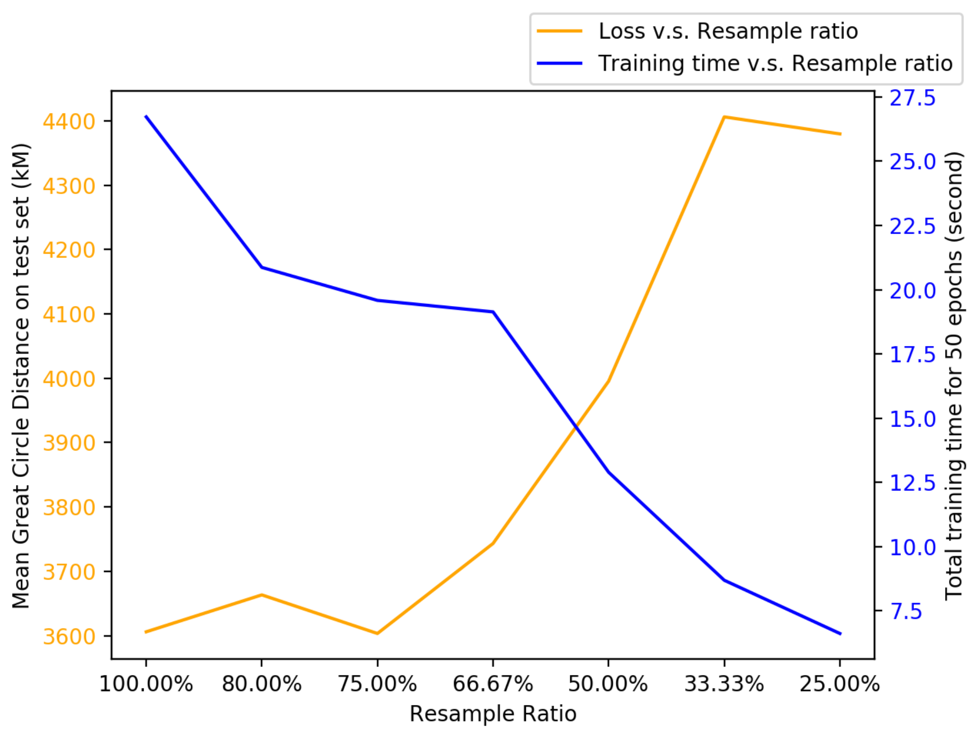

Fig. 2. Loss of Different Neural Network Trained with Training Set in Different Size

The diagram above demonstrates that with the resample ratio decreased (i.e., the size of new training set becomes smaller), the training time will also reduce. However, the loss will get increased at the same time. Therefore, we need a trade-off between the resample ratio and training time.

Figure 3 shows us the result of neural network whose features are selected by evolutionary algorithm. The small circles are outliers while the red broken line shows us the average of the individuals' errors in each epoch. The orange lines in each box represent the median of the individuals' errors in each iteration. Comparing to the result shown in Figure 1, after applying feature selection on our neural network model, the error is less than it of our simple solution. The improvement is more significant in our solution with evolutionary algorithm than that with heuristic pattern reduction. However, using evolutionary algorithm to decide which feature to be selected to improve the performance of neural network needs more time than using heuristic pattern reduction. Therefore, in practice, we need to consider the tradeoff between performance and time.

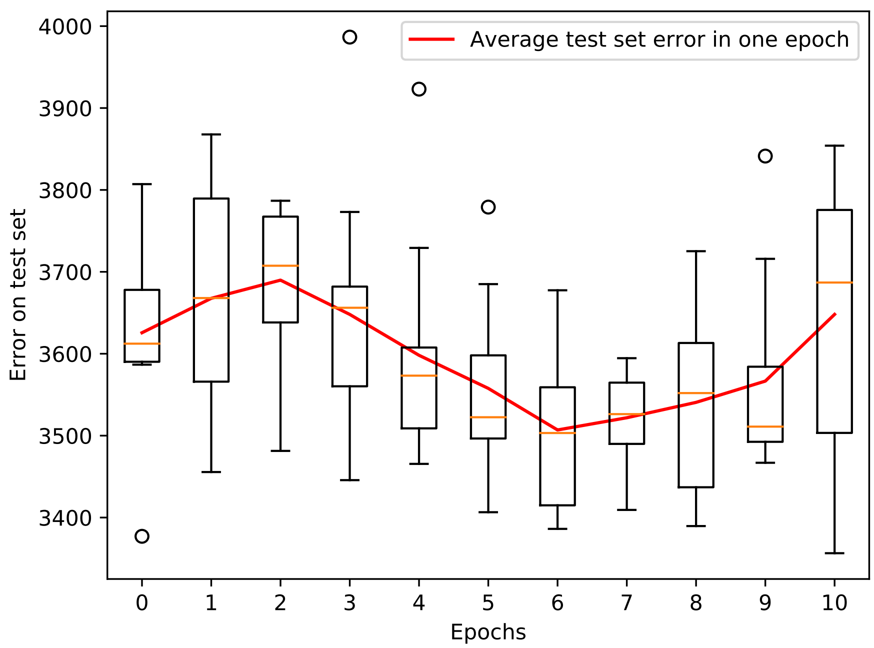

Fig. 3. Error of Individuals in Each Generation

3.3 Evaluation of Different Approaches

According to the diagrams above and the figures provided by [8], we can find that using the same dataset to solve the same problem, machine learning algorithms can get a better result. In [8], the minimal loss is less than 3200 while in our simple neural network solution, it is only minimized to around 3700. When applying theories of heuristic pattern reduction on the neural network, the minimal loss is around 3600. When utilizing evolutionary algorithm to do feature selection on input features of our neural network, the average of minimal loss is around 3500 while some individuals can get to a very close result as it in [8]. Though machine learning approaches perform better, a problem-specific similarity matrix needs to be defined in these methodologies, which makes the approach itself become more specific to the problem. On the other hand, neural network methodology doesn't need to do this kind of definition, which makes the methodology becomes more general. Also, if we do not consider the problem of time consuming, with feature selection done by evolutionary algorithm, we can get a result as good as the machine learning approaches with more generalisation.

3.4 Summary

In this chapter, we discuss and analysis the results we get from different approaches. We find that approaches based on machine learning algorithms perform much better than neural network. However, machine learning approaches are not as general as our approaches since they need to define the problem-specific similarity matrix. We also find that with same validation test set, reduce the size of training set and increase the learning rate correspondingly will help the neural network get a similar or even better result with less training time. Moreover, by using evolutionary algorithm to select features for input neurons will improve the performance of neural network.

4 Conclusion and Future Work

This is the last chapter of this report. We will draw a conclusion and discuss some work we can do in the future. In section 4.1, we will summarize all the content in this report and demonstrate what we have got. In section 4.2, future work of the predicting the origin of a piece of music problem will be discussed.

4.1 Conclusion

In this report, we mainly focus on how to solve the problem of predicting the origin of a piece of music. We implemented a 4-layer feed-forward neural network and trained it with back-propagation algorithm in stochastic gradient descent training model. Considering the weaknesses of back-propagation algorithm on training speed, we use heuristic pattern reduction to improve the performance of our neural network. We found that while the validation test set remains unchanged, the training speed will get increased with reduce the size of training set and increase the learning rate. Meanwhile, the result won't have significant change as well. We also use evolutionary algorithm to do feature selection. With this operation, we can filter some unnecessary features and only use the rest of features in the training and testing set to train our neural network. We found that while feature selection is a time consuming process, this methodology can significantly improve the performance of our neural network. We also compare our results gained by neural network to the solution based on other machine learning algorithms. Machine learning algorithms like KNN and Random Forest Regression can get much better result but these kinds of methodologies are not as general as neural network solutions.

4.2 Future Work

There is still some work for us to do to improve our neural network solutions. For heuristic pattern reduction solution, on one hand, in section 2.3, when we reduce the size of training set, percentage like 80% or 75% is chosen randomly. Therefore, we can find some self-adaptive functions to decide the percentage we apply on the original training set to reduce training patterns. On the other hand, according to the result we get, if we use 75% data of the full-size training set, the mean square error get a minima while with the size of training set keeps decreasing, the result become worse and worse but the training time keeps decrease. Thus, we need to find a trade-off between the accuracy and training speed. As for feature selection done by evolutionary algorithm, in the current solution, we treat each feature as an independent identity while in practice, there might exist some connections between features. Therefore, we can find some automatic methods to find out which features are relevant or which are irrelevant so that when we want to delete one feature, we can delete the features that related to it at same time.

References

1. Fraser, A., Burnell, D., et al.: Computer models in genetics. Computer models in genetics. (1970)
2. Gedeon, T., Bowden, T.: Heuristic pattern reduction. In: International Joint Conference on Neural Networks. vol. 2, pp. 449–453 (1992)
3. Gedeon, T., Harris, D.: Creating robust networks. In: Neural Networks, 1991. 1991 IEEE International Joint Conference on. pp. 2553–2557. IEEE (1991)
4. Kruschke, J.K.: Improving generalization in backpropagation networks with distributed bottleneck. In: Proc. Int. Joint Conf. on Neural Networks. pp. 443–447 (1989)
5. Morgan, N., Bourlard, H.: Generalization and parameter estimation in feedforward nets: Some experiments. In: Advances in neural information processing systems. pp. 630–637 (1990)
6. Setiono, R., Liu, H.: Neural-network feature selector. IEEE transactions on neural networks **8**(3), 654–662 (1997)
7. Tzanetakis, G., Cook, P.: Marsyas: A framework for audio analysis. Organised sound **4**(3), 169–175 (2000)
8. Zhou, F., Claire, Q., King, R.D.: Predicting the geographical origin of music. In: Data Mining (ICDM), 2014 IEEE International Conference on. pp. 1115–1120. IEEE (2014)