Project report for course Comp 766
Fall 2016

Prediction of Phylo's Puzzle Difficulty

Student:	Faizy Ahsan
Instructor:	Prof. Jerome Waldispuhl

School of Computer Science
Center for Bioinformatics
McGill University, Montreal
December 2016

Abstract

Phylo is a citizen science project for improving multiple sequence alignment (MSA). Throughout the years, Phylo has successfully engaged its user base for the MSA task through the puzzle gamification. It would be quite informative to know the quality of puzzles being solved on Phylo in terms of designing puzzles and gaining insights from the puzzle solutions. In this report, we propose a machine learning based methodology to assess the puzzles quality through its difficulty and to identify the significant puzzle features. We show that the learned model based on support vector regression can successfully predict the puzzle difficulty and random forests could identify its salient features.

1 Introduction

Citizen science project is a research collaboration of non-professional and professional scientists. Phylo is a citizen science project for improving multiple sequence alignment (MSA). In Bioinformatics, aligning sequences like DNA, RNA or protein is of great significance as it can explain the biology behind the functions associated with these sequences. Phylo presents MSA of various DNA sequences from different species as a visual grid puzzle. Each sequence is composed of character c, where c $\in$ { A, C, G,T, gap} and is represented as a row. The characters c are represented as colored blocks (four colors for four nucleotides and no color for gap). The sequences are depicted as leaves of a phylogenetic tree, which is provided adjacent to the puzzle. The user is required to align blocks column wise such that the resulting alignment is as close as possible to the true alignment.

The quality of sequences being aligned on Phylo can differ from each other in terms of the information they reveal. For e.g. some set of sequences could be easier to align, while others could be harder. The information about the characteristics of a puzzle that describe its difficulty level would be useful in designing Phylo puzzle. In this direction, we could learn a machine learning model to identify the difficulty level of a puzzle. In this report, we present the prediction of puzzle difficulty as a regression problem and identify the important characteristics for designing a puzzle.

The proposed regression problem is defined in §2. Our methods are described in §3. The data we use to learn the model is presented in §4 and we discuss our results in §5. Finally, we conclude with the analyses of our approaches and future work in §6.

2 Problem Definition

We define puzzle difficulty of a puzzle P as,

$$\text{puzzle-difficulty}[P] = \frac{\text{fail-count}}{\text{play-count}} \tag{1}$$

where play-count is the total number of time P has been played and fail-count is the number of unsuccessful attempts made to solve the puzzle. Our problem is to learn a model M that would predict the puzzle difficulty of a given P and to identify the salient features of P.

3 Methods

In this section, we describe the features identified from a puzzle and the learning model for the problem defined in §2.

3.1 Feature extraction

We consider the following features to represent a given puzzle,

- Number of sequences to align
- Number of stretches of gaps in the global alignment
- Mean square of the differences of pairwise length of the sequences to be aligned.
- Number of mismatches
- Mean of the evolutionary distances between the sequences
- Entropy of the nucleotides per column

3.2 Learning Model

We learn a support vector regression (SVR) [3] based model to solve the regression problem mentioned in §2. We use the scikit-learn package [2] to implement the SVR algorithm. There are three hyper-parameters for SVR namely, penalty cost to loss function (C), epsilon value of the ϵ-Loss function and the kernel coefficient of the kernel function. The ϵ-Loss function assign zero penalty to points lying in the ϵ tube. We use radial basis function, a gaussian kernel as the kernel function. We learn the hyper-parameter values on the validation set and we found the following suitable values for the three hyper-parameters,

- C: 46415
- kernel-coefficient: 2.15e-05
- ϵ: 1e-06

3.3 Feature Selection

We use ExtraTreeClassifiers algorithm based on random forest [1] from the scikit-learn package [2] to identify the salient features of the puzzle. We used the following command,

```
# fit an Extra Trees model to the data
model = ExtraTreesClassifier()

model.fit( x, y)

# display the relative importance of each attribute
print(model.feature_importances_)
```

The more important features in terms of learning the regression model would have the higher coefficient value.

4 Data

We utilized the Phylo platform data from 2012 to 2015. Phylo represents puzzle with lots of mysql tables. We used the table 'levels' to calculate the difficulty as defined in §2 and the table 'alignments' to extract the features mentioned in §3.1. In total, there were 1615 puzzles with play-count as non zero values. Figures 1 and 2 show the histograms and the scatter plot of difficulty level of these puzzles.

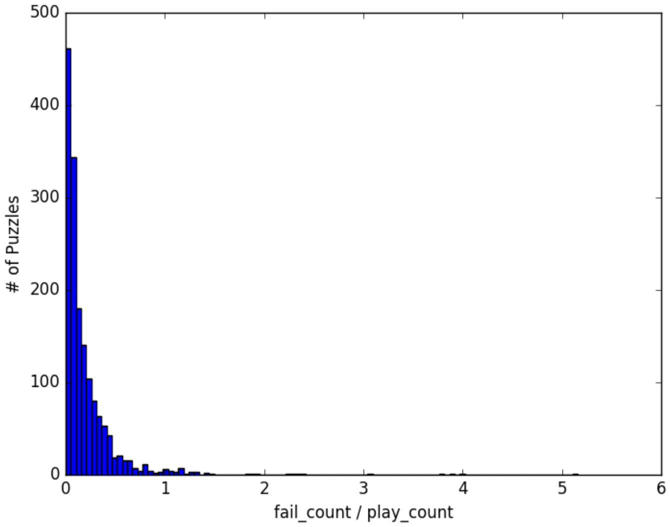

Figure 1: Histogram plot of the difficulty of the Phylo puzzles

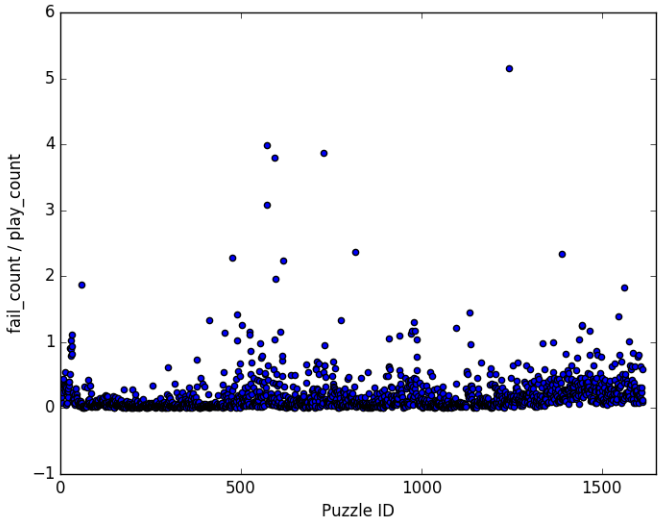

Figure 2: Scatter plot of the difficulty of the Phylo puzzles

5 Results

We use the R-squared (R^2) coefficient score to calculate the accuracy of the SVR model, which is defined as,

$$R^2 = (1 - \frac{u}{v}) \tag{2}$$

$$u = \sum_{i=1}^{N} (y_i - \hat{\mathrm{y}}_i)^2 \tag{3}$$

$$v = \sum_{i=1}^{N} (y_i - \mu)^2 \tag{4}$$

$$\mu = \frac{1}{N} \sum_{i=1}^{N} (y_i) \tag{5}$$

where N is the number of puzzles, y_i and $\hat{\mathrm{y}}_i$ are the true and predicted difficulty of puzzle i. The best possible value of R^2 could be 1.

We partitioned the data into two-third as train set and one-third as test set. We achieve the R^2 value of 0.99 (with C: 40000.0, kernel coefficient: 1.0 and ϵ: 0.01) and 0.254 (with C: 46415.0, kernel coefficient: 2.15e-05 and ϵ: 1e-06) on train and test set respectively. Using ExtraTreeClassifiers algorithm, we obtain the following coefficients for the features,

- a. Mean square of the differences of pairwise length of the sequences to be aligned: 0.23157225
- b. Number of mismatches: 0.2311426
- c. Mean of the evolutionary distances between the sequences: 0.23429225
- d. Number of sequences to align: 0.03592951
- e. Number of stretches of gaps in the global alignment: 0.1952873
- f. Entropy of the nucleotides per column: 0.07177609

This indicates that the features a, b, c are important in assigning difficulty to a puzzle.

6 Conclusion

We propose a machine learning based methodology to predict the difficulty level of Phylo's puzzle. We designed puzzle features that could determine the difficulty level and developed a method to measure the difficulty level. We identify that the pairwise length differences, mismatches and evolutionary distances of the sequences to be aligned play major role in determining puzzle's difficulty level.

However, the result on test set is not satisfactory. If we look at the distribution of difficulty level (Figures 1 and 2), it's clear that majority of puzzles have very low difficulty level. This indicates that a better criteria for determining puzzle's difficulty is needed or more data is needed to estimate the true difficulty distribution. Moreover, more careful feature designing could further improve the learning model. In particular, we should have considered a lot more features for the feature selection stage. With all these insights, future work would be significant in improving Phylo puzzle design.

References

[1] Tin Kam Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[3] Vladimir Vapnik, Steven E Golowich, Alex Smola, et al. Support vector method for function approximation, regression estimation, and signal processing. *Advances in neural information processing systems*, pages 281–287, 1997.