Classifying Tumor Samples with Genetic Mutations into Cancer Types by Machine Learning Techniques

Seongbeom Park
Student ID : 20165112
amita90@unist.ac.kr

Jinsu Park
Student ID : 20165126
jinsupark@unist.ac.kr

1 INTRODUCTION

Since there are various symptoms for the cancer, it is hard to know cancer before it becomes critical. To overcome this problem, there are various ways are developed to diagnose cancer with blood test [7, 9], because blood vessel is main carrier for the cancer metastasis to other organs or tissues.

Cancer type classifying is as important as cancer detection. To make tomography of a human body, there might be side effect due to radiation exposure with PET or CT scanning, and it is hard to discover small sized tumor with images before it developed.

This report addresses the way how we use machine learning techniques to classify the cancer type with DNA mutation of tumor samples. We use deep neural network (DNN) and light gradient boosting machine (LightGBM) [10] to build multi-label classifier. The model shows prediction accuracy up to 65.4% and we discussed about the way how to enhance the prediction accuracy of the model.

2 BACKGROUND

Machine learning has been emerged as a promising solution for complex problems such as computer vision, natural language processing, and robot locomotion. In machine learning, users train a model that produces desired output based with a large dataset. One of the advantages of machine learning is that it eliminates the need for explicit programming of the complex task.

Human DNA includes about 20,000 protein-coding genes [8] and it is slightly different to each other, because the gene can be varies by germline mutation (given from parents) and somatic mutation (caused by cell division). DNA Sequencing [5, 6, 12] is a technology to uncovers the sequence of a human. The speed of sequencing becomes faster by parallelize the process of copying DNA segment and analyze sample with high performance computers.

Genomics has recently introduced machine learning techniques, because human genome data is too huge to be analyzed by human. One of the applications is the cancer type prediction based on somatic point mutations data. DeepGene [13] is a state-of-the-art technique for cancer type classification based on deep learning and somatic point mutations.

In [13], the authors mention three major challenges in cancer type prediction – (1) only a small subset of genes is related to the cancer classification although genomic sequencing results include extremely large number of genes, (2) even within the small subset, the majority of genes are not related with mutations, which results in sparse gene data, and (3) the correlation between genes and cancer types are so complex that conventional linear classifiers cannot achieve high accuracy.

DeepGene achieves high accuracy by using the clustered gene filtering (CGF), the indexed sparsity reduction (ISR), and the DNN classifier.

3 DATA ANALYSIS

The provided dataset (TCGA_6_Cancer_Type_Mutation_List) has total 2284 tumor samples (i.e., Tumor_Sample_IDs), each of sample is classified into one of the six cancer types (i.e., BRCA, COADREAD, GBM, LUAD, OV, and UCEC). The number of samples of each cancer type is 977 (BRCA), 223 (COADREAD), 290 (GBM), 230 (LUAD), 316 (OV), and 248 (UCEC), respectively.

We randomly selected 457 samples from the entire dataset to generate the test dataset and the remaining samples are used for the training dataset. The ratio of samples included in each cancer type to the total samples is same in the entire, test, and training datasets.

Each sample has different number of mutations. For example, the first sample (i.e., TCGA-A1-A0SB) has 17 mutations and the second sample (i.e., TCGA-A1-A0SD) has 25 samples. To describe mutations of each sample, we use one-hot encoding for the gene list and perform reduction for the mutations of each sample. Each sam-

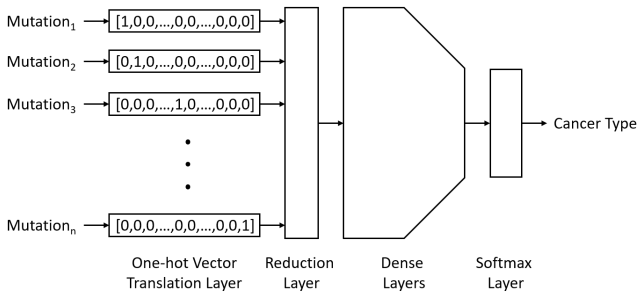

Figure 1: Custom DNN model overview

ple has a vector with 20743 (i.e., the total number of genes in the entire dataset) elements, each of element represent a gene. An element is marked as '1' if the sample has a mutation on the gene. Otherwise (i.e., the gene is not mutated), it is marked as '0'. We mark multiple mutations on the same gene as '1' and omit the mutation type information (e.g., variant type, reference allele, tumor allele) to avoid over-complicating the classification problem.

In summary, a tumor sample is transformed to which label is a cancer type and feature is a tensor that represents mutated genes.

4 DESIGN AND IMPLEMENTATION

We tried to build multi-label classifier with DNN and light gradient boosting machine (LGBM). This section describes how each model is built and how it works.

4.1 DNN

We implemented multi-label DNN classifier by using tensorflow API [3]. Though tensorflow provides predefined DNN classifier, we need to build a custom DNN classifier to train the dataset and evaluate the model. Since predefined DNN classifier is capable to use the input feature which has same structure. However, there can be different number of mutations for a cancer which cannot be used for predefined DNN classifier. There is a choice that preprocess input dataset which row is a tensor indicating multiple mutations for a sample, which column length is same with the number of human genes. The preprocessed dataset, however, takes too much volume of storage to maintain the dataset. We want to make a model which is scalable, but the dataset may cause out of memory to load whole dataset or low performance to load partial input data.

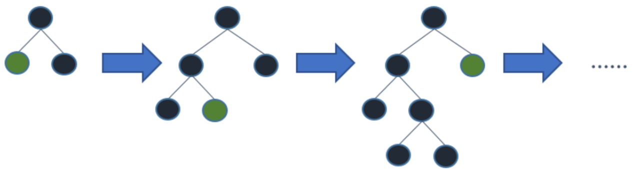

Figure 2: Leaf-wise tree growth of LightGBM [4]

Our custom DNN classifier consumes arbitrary number of mutations for a sample and predict the cancer type of the sample. Figure 1 shows the overview of the model. The input of the model is a batch which includes all mutations for a sample. Each mutation is translated to an one-hot vector in the ont-hot vector translation layer. The model reduces one-hot vectors to a single vector in the reduction layer indicating whether the gene had been mutated or not; 0 for normal gene, 1 for mutated gene. The reduced tensor becomes a input tensor for dense layers. We use fully connected neural network for 13 dense layers with 0.5 dropout. We use linear function for activation, because sigmoid and rectified linear unit (ReLU) functions converge too fast which causes overfitting problem.

The result of dense layers are converted to a tensor with 6 variables by using softmax function in the softmax layer. the softmax function calculates the portion of each tensor element and it is widely-used multi-label classifier. We selected the most highest possibility of cancer type with the output to calculate the accuracy of the model. We use sigmoid cross entropy to calculate loss which is an error between expected value with estimated value and we use ADADELTA [14] to optimize the model which changes the weights of dense layers in the direction of reducing the loss of the model.

4.2 LightGBM

We also implement a multiclass classifier by using LightGBM [10, 11]. LightGBM is a fast and high-performance gradient boosting framework based on decision tree algorithms, which is mainly used for machine-learning tasks including classification and ranking. It has been developed as a part of the Distrubuted Machine Learning Toolkit (DMLT) project [1] of Microsoft and recently showed outstanding achievments [2] in various machine learning challenges.

One of the important design characteristics of LightGBM is the use of the leaf-wise tree grows. Figure 2 and Figure 3 shows the tree growth of LightGBM and other boosting algorithms. While other algorithms grow tree horizontally (i.e., level-wise), LightGBM grows tree

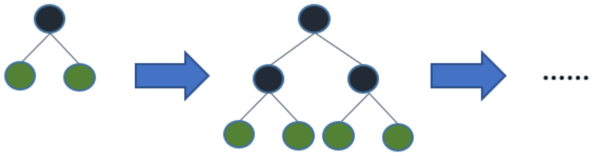

Figure 3: Level-wise tree growth of other boosting algorithms [4]

Table 1: LightGBM parameters

Parameter	Value
boosting_type	gbdt
objective	multiclass
num_class	6
metric	multi_logloss
num_leaves	255
min_data_in_leaf	200
max_depth	8
max_bin	255
num_leaves	255
subsample_for_bin	1000000
learning_rate	0.01
num_boost_round	5000

vertically (i.e., leaf-wise). As a result, LightGBM can achieve lower loss than other level-wise boosting algorithms for the same number of leaves.

Another important characteristic of LightGBM is Exclusive Feature Bundling (EFB). EFB is effective for reducing the number of features. In many real-world applications with a large number of features, the feature space are often sparse. The cancer type mutation list data used in this project also has a sparse feature space because we use one-hot encoding for the gene list to represent mutated genes (that is the reason why DeepGene [13] introduces the clustered gene filtering and indexed sparsity reducing techniques to improve the accuracy and speed of the classifier). LightGBM has an algorithm for merging exclusive features that can automatically reduces the number of features in the sparse feature space.

LightGBM provides many tuning parameters for the leaf-wise tree growth algorithm, including *num_leaves* (i.e., maximum number of leaves in one tree), *min_data_in_leaf* (i.e., minimal number of data in one leaf), and *max_depth* (i.e., limitation of the maximum depth for tree model). While LightGBM can converge much faster than other level-wise tree growth algorithms, the tuning parameters should be carefully determined to avoid over-fitting. Table 1 summarizes the baseline settings for the LightGBM parameters used in this project. We set the parameters on the basis of the parameter tuning guide, a multiclass classification example code provided by the official LightGBM code repository, and machine learning challenge winning solutions based on LightGBM.

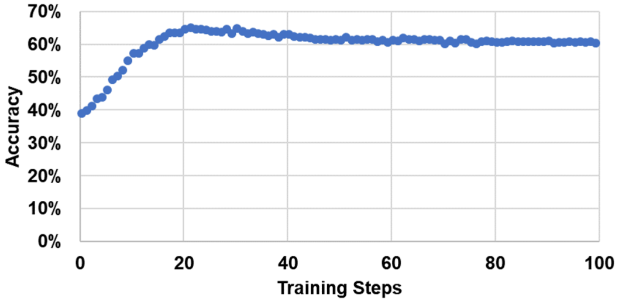

Figure 4: Overall accuracy with multiple training steps

Two important considerations of the parameter tuning are the trade-off between the training speed and the model accuracy and the over-fitting. We focus on the model accuracy rather than the training speed because LightGBM provides reasonable performance with the used dataset even when the performance-centric settings are used.

5 EVALUATION

The multi-label DNN classifier is trained and tested with randomly selected samples by using automated script which splits the dataset with 80% for training and 20% for test. We train the model 100 steps with training samples. For each training step, the training sample is shuffled randomly to protect the model learns biased due to the order of training samples.

Figure 4 shows the overall accuracy of the classifier while training the model multiple steps. The maximum accuracy is 65.4% when the training step is 22. The accuracy increases when the step is less than 22, because the model is underfitted. After trains more than 22 steps, accuracy decreases, because of overfitting.

Figure 5 shows the accuracy of each cancer type while training step increases. When the model is underfitted (i.e., less than 22 training steps), the model predicts cancer type for the sample as BRCA, because it is majority of dataset. Therefore, the accuracy for BRCA is nearly 100%, but the accuracy of other cancer types are small. As training step increases, the model predicts

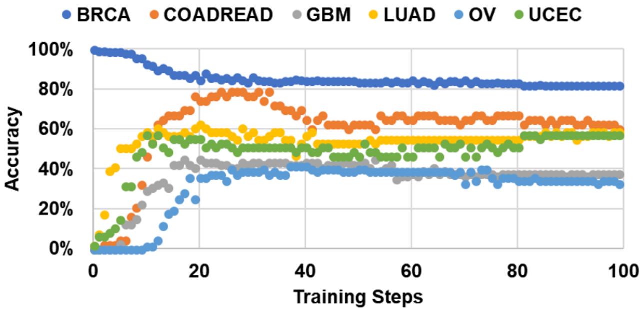

Figure 5: Accuracy for each cancer types

other cancer types with sacrificing accuracy of BRCA prediction. Whn the model is overfitted (i.e., more than 22 training steps), prediction accuracy for all types are decreasing tendency.

6 DISCUSSION

6.1 Enhance prediction accuracy

To simplify the problem for managing training and testing time, we used whether the gene is mutated as binary information. If variant type of each mutation is used for the model, prediction accuracy of the model can be enhanced.

If we had more dataset, we could use upsampling or downsampleing to balance the portion of each cancer type in the training dataset. It is widely-used method to samples in different frequency depends on the output label for the dataset which is biased. In the original dataset, there are more samples for BRCA. Therefore, the model has a tendency to predict cancer type to BRCA more than the other cancer types.

6.2 LightGBM accuracy

While LightGBM is a fast and accurate framework for classification problem, our cancer type classifier based on LightGBM is relatively inaccurate (i.e., classification accuracy for the test dataset is under 40%) compared to the DNN classifier. The LightGBM classifier shows 99.34% accuracy for the training dataset but tends to pick only one class for the test dataset.

We assume that the cause of the inaccuracy is the over-fitting due to the small dataset size. While the leaf-wise tree growth algorithm of LightGBM can achieve smaller loss than level-wise boosting algorithms for the same number of leaves, it is more vulnerable to the over-fitting. Although we have tried parameter tuning for the baseline parameter settings to avoid over-fitting, we have not been able to overcome the limitation of the small dataset size. Most of the machine learning challenges winning solution with LightGBM use dataset with more than 10K rows, while our dataset has only about 2K rows.

To solve the over-fitting problem, more advanced regularization techniques or larger dataset may be required.

7 CONCLUSIONS

In this project we build DNN and LGBM multi-label classifier to predict cancer type of tumor samples with arbitrary number of mutations with machine learning techniques. We described how we use machine learning to implement DNN and LGBM for multi-label classifier to predict cancer types, and evaluated the models. Our experimental result shows that prediction accuracy is up to 65.4%.

REFERENCES

[1] 2018. DMTK Project. https://github.com/microsoft/dmtk. (2018).

[2] 2018. LightGBM examples. (2018). Retrieved 2018-06-17 from https://github.com/Microsoft/LightGBM/blob/master/examples/README.md

[3] 2018. Tensorflow API (v1.4). (2018). Retrieved 2018-06-17 from https://www.tensorflow.org/versions/r1.4/api_docs/python/

[4] 2018. What is LightGBM, How to implement it? How to fine tune the parameters? https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc. (2018).

[5] Stephen Anderson. 1981. Shotgun DNA sequencing using cloned DNase I-generated fragments. *Nucleic acids research* 9, 13 (1981), 3015–3027.

[6] Wilhelm J Ansorge. 2009. Next-generation DNA sequencing techniques. *New biotechnology* 25, 4 (2009), 195–203.

[7] Joshua D Cohen, Lu Li, Yuxuan Wang, Christopher Thoburn, Bahman Afsari, Ludmila Danilova, Christopher Douville, Ammar A Javed, Fay Wong, Austin Mattox, and others. 2018. Detection and localization of surgically resectable cancers with a multi-analyte blood test. *Science* (2018), eaar3247.

[8] Iakes Ezkurdia, David Juan, Jose Manuel Rodriguez, Adam Frankish, Mark Diekhans, Jennifer Harrow, Jesus Vazquez, Alfonso Valencia, and Michael L Tress. 2014. Multiple evidence strands suggest that there may be as few as 19 000 human protein-coding genes. *Human molecular genetics* 23, 22 (2014), 5866–5878.

[9] Mark Kalinich and Daniel A Haber. 2018. Cancer detection: Seeking signals in blood. *Science* 359, 6378 (2018), 866–867.

[10] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A

highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*. 3149–3157.

[11] Qi Meng, Guolin Ke, Taifeng Wang, Wei Chen, Qiwei Ye, Zhi-Ming Ma, and Tieyan Liu. 2016. A communication-efficient parallel algorithm for decision tree. In *Advances in Neural Information Processing Systems*. 1279–1287.

[12] Frederick Sanger, Steven Nicklen, and Alan R Coulson. 1977. DNA sequencing with chain-terminating inhibitors. *Proceedings of the national academy of sciences* 74, 12 (1977), 5463–5467.

[13] Yuchen Yuan, Yi Shi, Changyang Li, Jinman Kim, Weidong Cai, Zeguang Han, and David Dagan Feng. 2016. DeepGene: an advanced cancer type classifier based on deep learning and somatic point mutations. *BMC bioinformatics* 17, 17 (2016), 476.

[14] Matthew D Zeiler. 2012. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* (2012).