Università degli Studi di Torino - M.Sc. in Stochastic and Data Science - A.Y. 2021/2022

Final project of Statistical Machine Learning (MAT0043)

Gene selection for cancer type classification

The purpose of our project was to work on a high-dimensional genomics data and find a relatively small number of genes to predict the cancer type of a given tumorous cells. This is known as Genes Selection for Cancer Classification and it is in line with many up-to-date problems of applied medicine.
Our dataset contained 1.032 cancerous cells and their knock-out probabilities, i.e. the probabilities of stopping the growth of tumor by inhibiting one of the ~ 17.000 genes. Each cell line was characterized by one of 10 possible cancer labels: Eye, Gastrointestinal, Gynecologic, Musculoskeletal, Neurological, Breast, Head-Neck, Blood, Genitourinary and, finally, Lung. We explored in details three Features Selection algorithms: Random Forests (RF) combined with Feature Importance, Lasso-SVM and Neural Networks (NN) combined with Olden Importance. We studied two Binary classification problems (Blood cancer vs Rest, Lung cancer vs Rest) and the Multiclass problem. Besides lung models, we achieved satisfying classification accuracies and we were able to select up tp 10% of genes. Models fitted on relevant variables obtained classification accuracies ranging from 67% to 99%.
Therefore, it seems that classifying cancer type from an extremely small set of genes depends on the cancer type itself. These methodologies worked incredibly well on Blood cancer, as we reached almost 100% accuracy with the reduced classifier, while failed miserably on Lung cancer.

1 Introduction

Cancer is a complex disease characterized by the uncontrolled growth of abnormal cells anywhere in the body. These abnormal cells are extremely invasive and we usually identify them with the name of their original tissue (for instance, breast cancer, lung cancer, brain cancer, etc.). In recent years, medicine has made a great step forward in finding new and efficient therapies for different diseases, including cancer. In particular, thanks to numerous advances in technology, collecting huge amount of data is no longer an issue, so that one can exploit information to define personalized treatments for patients. In this regards, the DepMap project[1] and, in particular, the Achilles project[2] aim to use genome-wide screens to collect data regarding mutations of cancerous cells, identify essential genes and report vulnerabilities across hundreds of human cancers.

Many researches are currently using DepMap datasets to identify a small number of genes which are responsible of cancers growth[3]. This procedure is often driven by medical knowledge, which we do not possess, together with some rough measures of importance. Being Maths student, we instead based our research on statistical models and on the hypothesis that "if a given classifier is able to distinguish different types of cancer, then the most relevant genes are the most important features for that classifier" (the meaning of "important" will be clarified later). Of course, selecting few truly significant genes has outstanding implications in the medical field: building faster diagnosis tools and synthesizing less toxic drugs are only two examples.

[1] DepMap Portal: https://depmap.org/portal/
[2] Achilles Project: https://depmap.org/portal/achilles/
[3] Background material: https://depmap.org/portal/publications/

2 Dataset

We used two public datasets from the DepMap Public 21Q3 database, released on August 2021[4]:

D1 *CRISPR_gene_dependency.csv*, containing 1.032 cancer cells and their 17.393 gene scoring results;

D2 *sample_info.csv*, containing cell lines information, such as primary disease and sample collection site.

Data were collected from real patients and successively processed, so that element (i, j) of this (1.032×17.393)-data frame is the probability that "knocking out gene j has a real depletion effect on the i-th cell". Before proceeding with our analysis, we removed missing values: only 10 rows coming from different tumours were involved. Regarding weird observations, we found 2 "Non-Cancerous" and 6 "Engineered" cells. The first can be reasonably discarded, whereas the latter requires a little care. Engineered cells are synthetically modified samples in lab and, here, they are manly associated to the Eye sample collection site. We decided to keep them and associate them to the cancer corresponding to their site.

We grouped the various cancer types in 10 classes according to common medical knowledge[5] and we obtained classes as reported in Figure 1. "Eye" was the smallest one as there were only 16 observations, 5 of which labelled as "Enginereed". On the other hand, "Gastrointestinal" was the largest group and it comprehended 7 types of cancer, making this group quite heterogeneous.

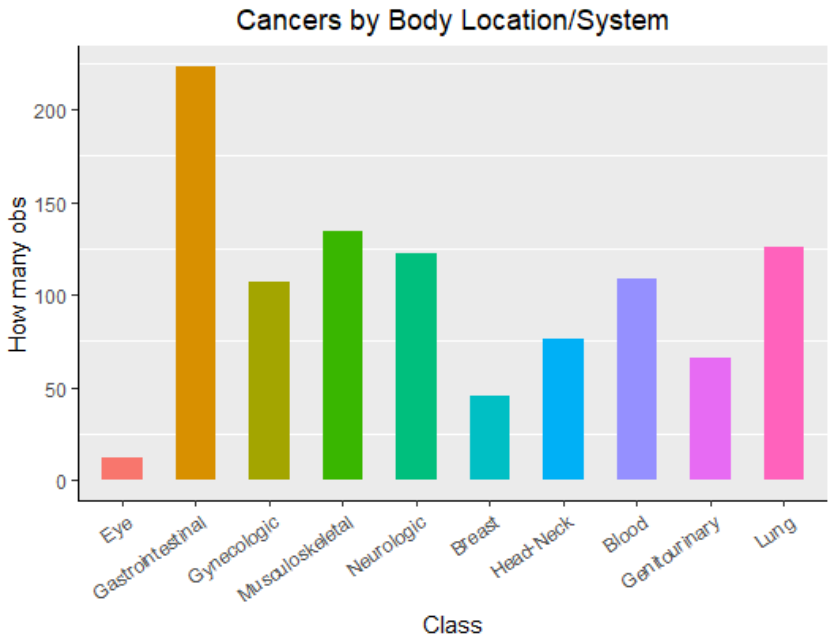

Figure 1: Cancer classes

We investigated two Binary classification problems, Blood vs Rest and Lung vs Rest, and the Multiclass problem. We chose "Lung" because of the nature of such a class: it was the most numerous group composed only by Lung cancer samples. The choice of "Blood" was instead driven by some underlying biological knowledge: Blood cancer is quite different from other tumours because

- Leukemia, Lymphom and Myeloma are the main kinds of cancer but they all affect white blood cells;
- blood is in the whole body, and so the cancer is, too,

3 Methods

Let us briefly illustrate the algorithms we used. Our methodology was characterized by three steps:

1. fit the model using all the features;
2. identify the most important variables based on some measure of importance;
3. use these genes to fit a reduced version of the classifier and find out its performance.

[4]Download dataset from DepMap: https://depmap.org/portal/download/

[5]Cancer types grouped by body location: https://www.cancer.gov/types/by-body-location

Clearly, each procedure involved fitting a model twice: the all-features version and the reduced one. We were thus forced to split the dataset into two further chunks. This was crucial to ensure independence of the two models and remove any sort of correlation.

3.1 Random Forest

We started by fitting Random Forest (RF) models as they frequently performs well on imbalanced and correlated high-dimensional data. We used Variable Importance to identify the most relevant features. This measure is calculated in three steps. First, prediction accuracy are measured on the out-of-bag samples. Then, the values of the variable are randomly shuffled, keeping all other variables the same. Finally, the decrease in prediction accuracy on the shuffled data is measured and the mean decrease in accuracy across all trees is reported. Intuitively, the random shuffling means that, on average, the shuffled variable has no predictive power.
Hence, Variable Importance measures how much accuracy decreases because of removing a variable. Here, we used it in two different ways:

- *Cross-validation*: we performed a 5-fold Cross-validation on the model, averaged the importance values and selected the top most important features;
- *Boruta algorithm*[6]: Boruta repeatedly measured feature importance and then performed statistical tests to screen out irrelevant features.

3.2 SVM-Lasso

Support vector machines (SVM) are based on the idea of finding a hyperplane that best separate classes. Here, we combined this method with the classical Lasso penalty, so that the objective function to be minimized was:

$$\frac{1}{n}\sum_{i=1}^{n} hingeLoss(y_i(x_i w + t)) + \lambda||w||_1 \qquad \text{where} \qquad hingeLoss(z) = max\{0, 1-z\}$$

The parameter λ has been chosen via cross-validation. Thanks to Lasso penalty, we obtained sparsity in predictors: some w_i were shrunken all the way to zero, whereas the others identified important features.

3.3 Neural Networks

Neural Networks (NN) are efficient models to capture non-linear relationships between predictors and target variables. In this context, we trained NN with two hidden layers of width $400/500$ and 300 respectively and we chose *ReLU* as activation function for the hidden layer, *sigmoid* and *softmax* for the output layer of, respectively, binary and multiclass classifications.
Given that the binary problems were a little unbalanced, we tried both the usual *Cross Entropy* loss function and the *Focal Loss*, defined as

$$FL(z) = \alpha \cdot (1-z)^{\gamma} \log z, \quad \text{with } z \in [0,1] \text{ and } \alpha, \gamma \geq 0$$

Note that *Focal Loss* can be extended for dealing with multiclass classification tasks[7].
Once our NNs were fitted, we ranked variables according to the Olden Importance measure[8], selected the first ones and trained a reduced version of the classifier on them. We used Olden's importance as it can work with multiple hidden layers and multiclass problems.

[6]Boruta algorithm: https://www.researchgate.net/publication/220443685_Boruta_-_A_System_for_Feature_Selection
[7]Focal Loss: https://arxiv.org/pdf/1708.02002.pdf
[8]Olden Importance: https://depts.washington.edu/oldenlab/wordpress/wp-content/uploads/2013/03/EcologicalModelling_2004.pdf

4 Results

Before starting our Binary classifications on Blood and Lung cancer, we ran a Principal Component Analysis to gain some valuable insights. We noticed a surprising result: even if observations formed a cloud of points, Blood cancer cells were mainly concentrated in just one part of the 3-dimensional plot. The same did not happen for Lung cancer observations.

4.1 Binary classifications: Blood vs Rest

We obtained remarkable results in Blood vs Rest classification. We initially fit a **Random Forest** (RF) classifier. Even though Blood cancer observations were only the 11% of the total, we did not need any adjustments for the minority class. Indeed, thanks to proper tuning on trees parameters and a correction on class weights, we reached 98% of mean accuracy. For the sake of completeness, we fit also a RF with Cost-Complexity Pruning and we find the same accuracy. We then removed unnecessary features by performing a 5-fold Cross-validation. Thanks to Variable Importance values, we were able to select the most relevant genes for each of the 5 models. As more rigorous way to pursue variable selection, we applied Boruta algorithm, too. As shown in Table 2, Boruta individuated a higher number of important features than our manual method and, in particular, they agreed only on 84 genes. As mentioned above, we fitted RFs on new reduced dataset, where cancer instances accounted by 9% of the total. Besides weighting classes, no further parameters were specified and, nevertheless, both RF classifiers predicted correctly 6 tumour cells out of 7.

As second model, we explored **SVM-Lasso** in the way it was described in the previous section. Being an embedded Feature Selection method (i.e. the learning algorithm intrinsically performs feature selection), fitting the model already provided 108 important genes, which are the ones associated with a non-zero weight. The average recall is now of 97%. Furthermore, 12 important features were shared with the RF model, which made us think that those genes might be of medical importance for real.

Furthermore, **Neural Network** (NN) classifier achieved outstanding performance as well. In this case, instead of tuning the NN's parameters in order to take into account the Blood minority class, we decided to fit 50 NNs (with the topology described in section 3.3) on 50 different undersampled dataset and then take the average to make predictions. Each of these fifty datasets was constructed by retaining all the Blood observations and randomly picking as many Non-Blood observations. The accuracy we achieved was almost 100%, hence we decided to select the most important genes according to Olden's importance. Since NNs are especially suitable in handling high-dimensional data, decided to keep more features than RF and try the simplest NN model, i.e. built as a "vanilla" neural net with one single hidden layer. As a result, we reached an accuracy of 98.8% for this reduced model, with all Blood cancer cells classified correctly.

4.2 Binary classifications: Lung vs Rest

Unfortunately, we could not build a proper model for Lung vs Rest. Even after a tuning of the tree parameters, the **RF** was not able to detected the minority class. Thanks to SMOTE, we adjusted the frequency of those observations from 11% to 50% and the refitted RF gave us a slightly better result: at least one third of Lung cancer cells was correctly predicted. Similar results were achieved with Minimal Cost-Complexity Pruning. In any case, we reported the average recalls in Table 1. Poorly results were found for **NNs** as well, even using the *Focal Loss*, and for **SVM-Lasso**. Since all the models were not good enough in classifying Lung cancer, we hypothesized that selecting the most important features was pointless and incorrect in terms of real applications hence no reduced model was fit.

Table 1: Average recall of the models fitted with all the features

Task	RF	NN
Blood	0.977	0.986
Lung	0.649	0.632
Multiclass	0.655	0.702

Table 2: Selected variables

Task	RF	Boruta	SVM-Lasso	NN
Blood	109	118	108	300
Multiclass	645	41	—	1700

Table 3: Average recall of reduced models

Task	RF	SVM-Lasso	NN
Blood	0.929	—	0.993
Multiclass	0.525	—	0.494

4.3 Multiclass classification

When working with the multiclass problem we decided to remove the group "Eye" because it was a small heterogeneous group that even after tuning for RF and NN held the worst accuracy scores. "Eye" was a cluster of different types of cancer: this didn't create problems when we were using a One vs All aproach but with multiclass the effects of doing the cluster ourselves was more evident. So after removing the group "Eye", we did the same as before and split the dataset in two.

Let us consider the RF classifier. At first we fitted two of them, the former with balanced weights and the latter balancing the weight by looking at each class ratio. This was done because if we look at the confusion matrix we can see hat most of the missclassification errors are data points classified as Gastrointestinal. Gastrointestinal was the biggest cluster we consider with a lot of different cancers so the genes(i.e. the features) important can be different within the cluster. To limit the weight of Gastrointestinal we decided to tune by hand the class weights. At this point we used cross validation to select five models and we extracted the most important features by looking at their relative importance in every model. We set a threshold to obtain 465 features and with this we created a reduced model that we fitted using the validation set we had created at the beginning. At last we used Boruta algorithm to find a different set of important variables and we fitted a RF to compare it with the other reduced model.

Then we moved on NN. We then fitted three different NN: the first increasing by one the number of hidden layers, the second changing the loss function with the Focal loss and the third using again the Focal loss, but changing the alpha parameter w.r.t. the class percentage in the set. The three models held similar result with the second one being slightly more accurate. Once this NNs were fitted, we ranked the variable according to the Olden Importance measure, selected the first 1700 and used them to construct a reduced version of the classifier on the validation set. The increment in the number of feature used for fitting the reduced model w.r.t. the binary NN model was beacuse we are now trying to separate nine different classes so we need much more information than before.

5 Conclusion and future works

- reiterate key findings

- limitations: data high correlated, no medical knowledge, no meaning at selected genes
- suggest for the future: work with a med student, try other classes or methods, improve lung in some way

We approached the multiclass problem using three different techniques: RF, NN, SVM-Lasso and all three methods gave us good results. We can't say the same for the reduced models: while NN maintain an overall good performance using the same models, the RF need to be retuned to reach a similar performance.