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 is 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 contains 1.032 cancerous cells and their knock-out probabilities, i.e. the probabilities of stopping the growth of tumor by inhibiting one of the $\sim$ 17.000 genes. Each cell line is characterized by one of 10 possible cancer labels: Eye, Gastrointestinal, Gynecologic, Musculoskeletal, Neurological, Breast, Head-Neck, Blood, Genitourinary and, finally, Lung. We explor in details three Features Selection algorithms: Random Forests (RF) combined with Feature Importance, Lasso-SVM and Neural Networks (NN) combined with Olden Importance. We study two Binary classification problems (Blood cancer vs Rest, Lung cancer vs Rest) and the Multiclass problem. Besides Lung models, we can achieve satisfying classification accuracies and we are able to select up to 10% of genes. Models fitted only on relevant variables obtain good classification accuracies, too.
Our methodologies have worked incredibly well on Blood cancer, as we have reached almost 100% accuracy with the reduced classifier, but failed miserably on Lung cancer. Therefore, it seems that classifying cancer type from an extremely small set of genes depends on the cancer type itself.

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. 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 use 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 remove missing values: only 10 rows coming from different tumours were involved.

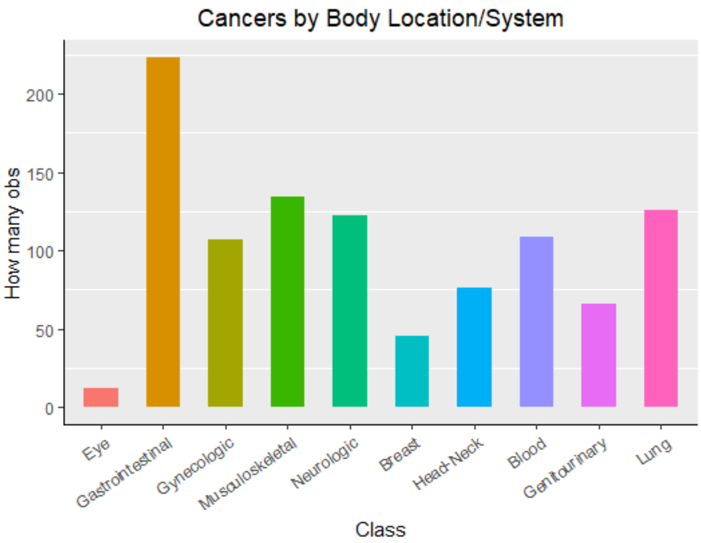

Figure 1: Cancer classes

By grouping the various cancer types in 10 classes according to common medical knowledge[5], we obtain the classes reported in Figure 1.
"Eye" is the smallest one as there were only 16 observations, 5 of which labelled as "Enginereed", i.e. synthetically modified samples. On the other hand, "Gastrointestinal" is the largest group and it comprehends 7 kinds of cancer, making this group quite heterogeneous.
We investigate two Binary classification problems, Blood vs Rest and Lung vs Rest, and the Multiclass problem. We choose "Lung" because of the nature of such a class: it was the most numerous group composed by only one type of cancer samples. The choice of "Blood" is 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 is 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.

Clearly, each procedure involves fitting a model twice: the all-features version and the reduced one. We are thus forced to split the dataset into two further chunks. Note that this is crucial to ensure independence of the two models and remove any sort of correlation.

[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

3.1 Random Forest

We start by fitting Random Forest (RF) models as they frequently performs well on imbalanced and correlated high-dimensional data. We use 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 variable removals. Here, we exploit it in two different ways:

- *Cross-validation*: we perform a 5-fold Cross-validation on the model, average the importance values and select the top most important features;
- *Boruta algorithm*[6]: Boruta repeatedly measures feature importance and then performes 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 combine this method with the classical Lasso penalty, so that the objective function to be minimized is:

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

The parameter λ has been chosen via cross-validation. Thanks to Lasso penalty, we gain sparsity in predictors: some w_i are shrunken all the way to zero, whereas the others identify 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 train NN with two hidden layers of width $400/500$ and 300 and we choose *ReLU* as activation function for the hidden layer, *sigmoid* and *softmax* for the output layer of, respectively, Binary and Multiclass classifications.
Since the Binary problems are a little unbalanced, we try both the usual *Cross Entropy* loss function and the *Focal Loss*, defined as

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

Notice that *Focal Loss* can be extended for dealing with Multiclass classification tasks[7].
Once our NNs have been fitted, we rank variables according to the Olden Importance measure[8], select the first ones and train a reduced version of the classifier on them. We use Olden's importance as it can work with multiple hidden layers and multiclass problems.

4 Results

In the next two subsections we present the outcomes obtained by fitting the models discussed above. These results are expressed in terms of average recall, i.e. $\frac{1}{k}\sum_{i=0}^{k} r_i$, where r_i is the rate of correct predictions for class

[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

i and k is the total number of classes. We prefer not to rely on the usual accuracy measure, which is defined as *total correct prediction / number of observations*, because it conveys to a misleading message. Indeed, we have reached 90% accuracy in the Lung vs Rest classification. However, our classifiers completely miss Lung cancer instances, which correspond to the 10% of the total, and so they are pretty useless. And, in fact, their average recall is 50%.

4.1 Binary classifications: Blood vs Rest

We obtain remarkable results in Blood vs Rest classification. We initially fit a **Random Forest** (RF) classifier. Even though Blood cancer observations are only the 11% of the total, we do not need any adjustments for the minority class. Indeed, thanks to proper tuning on trees parameters and a correction on class weights, we reach 98% of average recall, see Table 1. For the sake of completeness, we fit also a RF with Cost-Complexity Pruning and we find the same result. Then, we focus on feature selection using both RF Variable Importance and the Boruta algorithm. As shown in Table 2, Boruta individuates a higher number of important features than our manual method and, in particular, they agree only on 84 genes. As mentioned above, we fit two RFs on a second dataset, one for each set of selected features. Besides weighting classes, no further parameters are specified and, nevertheless, both RF classifiers predict correctly 6 tumour cells out of 7, with an average recall of 93%.

As second model, we explore **SVM-Lasso**. Being an embedded Feature Selection method (i.e. the learning algorithm intrinsically performs feature selection), the model already provides 108 important genes, which are the ones associated with a non-zero weight. The average recall is now of 97%, as reported in 3.

Furthermore, **Neural Network** (NN) classifier achieves 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 decide to fit 20 NNs on 20 different undersampled datasets and then take the mean class probability to make predictions. Such datasets are constructed by retaining all the Blood observations and randomly picking as many Non-Blood observations. Having reached a high average recall, namely 98.6%, we select the most important genes according to Olden's importance. Since NNs are especially suitable in handling high-dimensional data, we decide 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 gain an average recall of 99.3% and all Blood cancer cells correctly classified.

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** is not able to detect the minority class. Thanks to SMOTE, we can adjust the frequency of those observations from 11% to 50% and now the refitted RF gives us a slightly better result: at least one third of Lung cancer cells are correctly predicted. Similar results are achieved with Minimal Cost-Complexity Pruning. Poorly results have been found for **NNs** as well, even using the *Focal Loss*, and for **SVM-Lasso**. We report all average recalls in Table 1. Since all the models are not good enough in classifying Lung cancer, we retain that selecting the most important features is pointless and incorrect in terms of real applications. Therefore, no reduced model has been fitted.

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

Table 3: Average recall of reduced models

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

4.3 Multiclass classification

When working with the Multiclass problem, we decide to remove the group "Eye" as it is too small and heterogeneous, thus extremely difficult to detect by the classifier. Besides that, we still split the dataset into two chunks, one for finding the important genes and the other one for fitting and testing reduced classifiers, and proceed in a similar fashion as above.

Let us consider the **RF** classifier. We start by fitting it with balanced weights classes. However, by looking at the confusion matrix, we spot most of the missclassification errors on Gastrointestinal class. Gastrointestinal is the biggest cluster and it is made of many different cancers. Thus, we expect many genes are involved to different this group. We then use 5-fold Cross validation to extract the most important features by looking at their relative importance. With our threshold on the importance, we obtain 465 features and fit a new RF on the reduced dataset. Here, the manual tuning on class weights to limit the effect of Gastrointestinal gives a better result than the balanced class weights. At last, we also exploit Boruta algorithm and we find way less important variables than with Cross-validation, as one can see in 2, but the two RFs achieve a similar average recall.

As before, we study **SVM-Lasso** as second model. Although it is particularly suited for binary classification tasks, we extend it to the Multiclass problem by using the so-called OVO (One vs One) approach. In other words, we take every possible combination of two classes and, for each of them, we construct an SVM-Lasso model. We then end up with $\binom{9}{2} = 36$ models so that predictions are determined by seeing which class wins more "duels". Again, being an embedded model, feature selection is already achieved by fitting the model once and all outcomes are displayed in Tables 2 and 3.

Finally, **NNs**. We fit three different NNs: one using the *CrossEntropyLoss* and the other two with *Focal loss*, but changing the α parameter w.r.t. class percentages. In particular, these last models are slightly more accurate. We take the best one to rank variables according to their Olden Importance measure. Here, we take more genes than for respective NN for binary classification because separating 9 classes requires more information. The average recall of the new classifier is shown in Table 3.

5 Conclusion and future works

We can conclude that classifying cancer type from an extremely small set of genes heavy rely on the cancer type itself: our classifiers are able to distinguish Blood cancer almost perfectly but not Lung Cancer. Notice that this thesis is also supported by the results on the multiclass task, as a significant decrease of the average recall is due to bad-behaved classes such as Lung (again!) and Gastrointestinal. Thus, a future study could be solve this issue. From a more general perspective, we admit that we do not have any hint about the meaning of selected genes and what is their role in the DNA. Hence, it could be interesting to involve Med students into a similar project and base our work on a more solid medical knowledge.