Structural Bioinformatic Project

Caregari Alberto Rapicavoli Fabiana Camuz Can Abdullah Rasouli Sina

July 17, 2022

1 Introduction

The aim of this project is to correctly predict the contact types in given protein structures, which are to be presented in PDB file format. To achieve this, a classification software is needed to calculate the probabilities of different contact types, which the residue pairs might have. The contact types that residue pairs in the protein structure can have are as follows:

- hydrogen bonds (HBOND);
- Van der Waals interactions (VDW);
- disulfide bridges (SBOND);
- salt bridges (IONIC);
- π-π-stacking (PIPISTACK);
- π-cation (PICATION).

Note that, if the software is unable decide (classify), the contact type between two pairs, it labels that as *Missing*.

In this project, different models were employed to perform the task of classification, the results were evaluated with different metric scores and best model was selected after comparison.

2 Description of the data-set

The data-set is composed by 1807 PDB files, in other words 1807 proteins coded in a global standard. Those files are supposed to be used for training our machine learning methods. We were also provide by a software that can achieve some feature from the PDB files, those feature later are used to teach how to recognize the type of bound between residues. The input data are composed by 34 columns, the first one is the PDB identity, the last one is the type of interaction (the label), half of the remaining columns are feature of source residue and the other half are of the target one. Those 17 features describe completely the residues: the first 5 columns identify the residue, the other 12 describe some geometrical (like ψ and ϕ angles) and chemical (like Atcheley features that correspond to polarity, secondary structure, molecular volume, codon diversity and electrostatic charge) properties of each residue. So, basically, we started from PDB files and obtained a table with n columns and m rows, labelled with the type of contact between residues. Here it is a bar-plot showing the distribution of contacts:

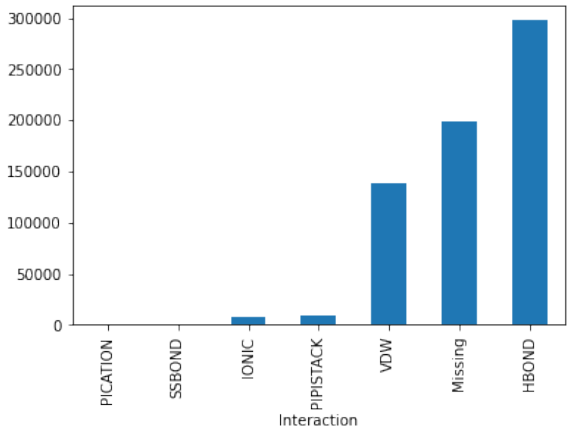

Figure 1: Picture of histogram

As we can see in Figure 1 the classes are strongly unbalanced, the number of *HBOND* in 3 order of magnitude bigger than the number of *PIPISTACK*. This situation of highly unbalanced data-set will lead us to choosing a binary classification rather than a multi classification, but

we will go deeper later.

3 Processing of data

Exploring the data, we noticed that some rows that has missing values in it, so, in order to prepare the data for the algorithms, those data were deleted, reducing from $735k$ to $652k$ rows. We had other options, in order to deal with missing values, like replacing the values with the mean of the feature of the column restricted to the same labelled rows, but this practise is somehow adding information, that are not true and it could cause wrong estimation in the prediction. We decided to delete the row because of the magnitude of the data-set: we have a huge number of observations and deleting a small percentage of them would not have such an effort in the final result. Since, to achieve our goal, we want to use machine learning algorithms, that use only numerical inputs, we deleted all the non-numerical features (for example we drop the `pdb_id`, `s_resn` and `t_resn` columns) After this, it was observed (see Figure 2) that the data had some features that were highly correlated to each other (more than 80%) so with this threshold, it was indicated that features: `s_up` ,`s_a5`, `t_a5` were highly correlated with `s_rsa`, `s_a3`, `t_a3` respectively. Then the first series of features were deleted and the features were reduced to 17 features from 20 .

4 Model training, performances and evaluation

As indicated above, the data have decent amount of samples for *HBOND*, *VDW* and *MISSING* label, but, for the other categories, there are not enough examples and algorithm may ignore them in learning process so for that matter. It is decided by writers to use under-sampling method to make a balance between the samples.
Over-sampling was not an option according to the reasons that are:

- The data-set is huge, so doing over-sampling is not an optimised way, from time and storage perspective.
- The gap between the categories is really high, that if oversampling was applied it would generate multiples copies of the same data, leading to over-fitting.

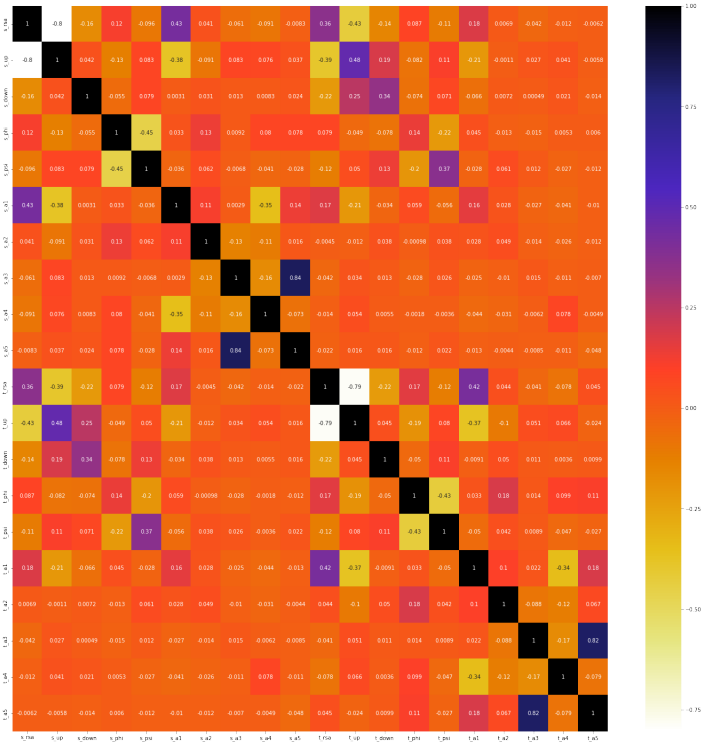

Figure 2: Heat map of features correlation.

So under-sampling was applied only when IONIC, PIPISTACK, SSBOND and PICATION bonds want to be identified, instead the data has been left as they were for HBOND, VDW, and MISSING.

Three classical machine learning algorithms are been implemented: Neural Network (both multi-class NN and binary NN), Random Forest (RF) and Support Vector Machine (SVM). Each of those algorithms has been built for binary classification, so the same algorithms was fitted with seven types of data, one for every type of bond. Label 1 is assigned if the bond belong to the class that is wanted to be identifier and 0 if the bound belong to one of the other six classes.

4.1 Neural Network

The model was created using `Keras` library from `Tensorflow` package. The model is composed by:

- In-punt layer: with the dimension of the feature after the pre-processing.
- Two hidden layers: the dimensions were reduced from seventeen to ten.
- Output layer: the dimension of the output is one with the *sigmoid* activation function.

As said before,the same model was fitted with seven different labelled data, for each it is made a ten fold validation. The result are shown in table 4.1. In order to show how significant is this approach to the problem, it was implemented a multi-class neural network, based on the previous model (only the last two layers has been changed with the aim of have an output of dimension seven and the softmax activation function) and the results were much less encouraging (54.2% of accuracy over all).

Bond	F1	Accuracy
HBOND	0.640	69.4%
VANDW	0.012	79.0%
IONIC	0.896	94%
PIPIS	0.936	95.5%
SSBOND	0.976	98.7%
PICAT	0.746	84.4%
MISS	0.319	71.1%

Table of scores for binary Neural Network.

4.2 Random forest

For the random forest, it has been decided to implement the binary classification with a grid search method for choosing the best combination of parameters. In particular, three different parameters for `n_estimators` and four for `max_depth` were chosen to be explored: respectively $5, 10, 20$ and $3, 5, 10$ and the default option `None`. The criterion used for the division at each node is based on the entropy. As before, a ten fold fold validation is used to calculate the F1 and accuracy score. The results can be seen in the table below.

Bond	F1	Accuracy
HBOND	0.702	71.3%
VANDW	0.012	79.0%
IONIC	0.971	97.1%
PIPIS	0.986	98.5%
SSBOND	1.0	100.0%
PICAT	0.977	97.8%
MISS	0.635	73.4%

Table of scores for binary Random Forest.

4.3 Support vector machine

The last machine learning algorithm used is the Support Vector Machine. In this case, it was not applied any grid search method, so all parameters were set as default. The Linear Support Vector Machine is preferred, rather than the classical when there are data-set of notable dimensions. It follows a table with the F1 and accuracy scores from a ten fold validation.

Bond	F1	Accuracy
HBOND	0.666	66%
VANDW	0.443	78%
IONIC	0.497	98%
PIPIS	0.496	98%
SSBOND	0.5	99%
PICAT	0.5	99%
MISS	0.523	67%

Table of scores for binary Linear Support Machine.

5 Conclusions

To sum up, four different approaches were selected to be performed on the data-set. Since the data was not balanced, it was decided to use under-sampling, instead of over-sampling, according to what has been said before. Random under-sampler method was called to obtain the balance data-set. According to what is indicated in the given tables about accuracy and F1-score for binary algorithms, it is shown that the random forest algorithm performs slightly better than the others, followed by NN, SVM and multi-class NN. The average accuracy of RF, NN, and SVM is: 88.16%, 84.59%, 86.43%, and 54.2% accordingly.