Mathematical Underpinnings of Machine Learning
Project A - Report

Grzegorz Zakrzewski, 313555

1 Introduction

The selected project is Project A - feature selection, which I worked on alone. The project involved comparison of feature selection methods. Two of them had to be based on mutual information, so I selected *Joint Mutual Information* and *Conditional Mutual Info Maximisation*. The other two feature selection methods I selected arbitrarily: one based on Random Forest feature importance and the other utilizing sequential feature selection. To compare these feature selection methods, I prepared three artificial datasets and found and downloaded three real-world datasets. The experiments were performed, and the reliability, effectiveness, and stability of the feature selection methods were evaluated.

2 Methodology

In this section, I describe the encompassed feature selection methods along with their stopping rules, enumerate the datasets used in the experiments, and discuss the results assessment.

2.1 Feature selection methods

Joint Mutual Information & Conditional Mutual Info Maximisation

Brown et al. [2] provides formulas for *Joint Mutual Information* (JMI) and *Conditional Mutual Info Maximisation* (CMIM), along with some useful transformations that were very helpful in implementation. I do not repeat these formulas here and do not provide a theoretical introduction to information theory due to report length restrictions.

JMI and CMIM select features in a step-by-step manner, so there was a need to devise a stopping rule. I came up with an idea that incorporates two conditions, which rely on configurable parameters. The first condition is a hard limit on the maximum number of features. If the feature selection method reaches this number, the process stops immediately. The default value is twice the square root of the total number of features. The second condition involves the JMI or CMIM criterion value. If the value in the current step is lower than a specific percentage of the average of the values from the two previous steps, the feature selection process stops. This percentage threshold defaults to 90% and was arbitrarily selected after a few quick experiments.

Random Forest feature importance

In the instructions, decision tree-based feature importance was proposed as a possible choice for one of the arbitrary methods. I decided to use a Random Forest model to obtain feature importance. In my setting, the Random Forest model is obtained after a small grid search over the `max_depth`, `min_samples_split`, and `min_samples_leaf` parameters. The number of estimators is always set to 50. Random Forest models are scored with 4-fold cross-validation, and the feature importance vector of the best Random Forest model is used. The maximum number of features is set in the same manner as in the case of JMI and CMIM. There is a second condition: the feature importance of every selected variable must be higher

than $\frac{1}{2 \cdot n\_features}$, since the values of *feature_importances_* in the `scikit-learn` package sum up to 1. The variables with the highest feature importance that meet these two conditions are selected.

Sequential Feature Selection

The second arbitrary method was prepared using the `SequentialFeatureSelector` class from the `scikit-learn` package. It is a recursive, wrapper, forward-selection method, which in a greedy fashion at each step adds a feature based on the cross-validation score of a selected estimator. I selected a simple nearest neighbor classifier ($k = 3$) to serve as my estimator. The maximum number of selected features was set as in the previous cases.

2.2 Datasets

Artificial datasets

I have prepared three different datasets to serve as input during the experiments. All three datasets have $p = 10$ features and $n = 500$ observations. The datasets are not large, and as a result, the computation did not take much time.

In the first dataset, all variables follow a normal distribution $X_i \sim \mathcal{N}(0, 1)$ for $i = 1, \ldots, 10$. The target variable is created by summing up the first five features and then dividing them into two equally sized bins 0 and 1 (based on the median of the obtained sum) using the `pandas` function `qcut`. The last five features take no part in the creation of the target variable, so they are completely irrelevant. The first artificial dataset will be called *irrelevant*.

In the second dataset, explanatory variables are created in the same way as in the first dataset. The difference is that the feature X_1 is correlated with feature X_6 with a correlation ratio of 0.9, feature X_2 is correlated with feature X_7 with a correlation ratio of 0.8, and so on, up to features X_5 and X_{10} correlated with a ratio of 0.5. To sum up, the first five features are relevant, while the last five features take no part in the creation of the target variable but are correlated with relevant features. The second artificial dataset will be called *correlated*.

The third artificial dataset is different. All variables follow a Bernoulli distribution with a probability of single success $p = 0.5$. The target variable is computed as the result of an XOR operation on the first two variables: $Y = XOR(X_1, X_2)$. All other variables are irrelevant. Mutual information-based feature selection methods should fail with this dataset because they will miss the more complex relationship between X_1 and X_2 that defines Y. The third artificial dataset will be called *XOR*.

Real-world datasets

I found and utilized three real-word data examples:

- *Abalone* - 4,177 rows, 8 features, transformed into binary classification task (source);
- *Mushroom* - 58,598 rows, 12 features (subset of 20 original features), binary classification (source);
- *Students dropout* - 4,424 rows, 36 features, classification to three classes (source).

I didn't want to use larger datasets because of computational reasons. However, within these three datasets, there are different characteristics - the second dataset (Mushroom) has a larger row-to-feature ratio, and the third one (Students) has this ratio lower.

2.3 Results assessment

Each experiment was repeated multiple times - artificial datasets were regenerated, and bootstrap samples were drawn from real-world data examples. I had three methods to assess the results of conducted experiments.

The first method to check and compare results was by fitting a generic classification model to a selected subset of features and measuring its performance. Following Brown et al. [2], I employed a simple nearest neighbor classifier ($k = 3$) for this task. Additionally, a classification measure was computed for a model trained on the full set of features. My classification measure was accuracy averaged for test folds in 4-fold cross-validation.

The second assessment method, applicable only to artificial datasets where relevant features were known, was the *Success Rate* proposed by Bolón-Canedo et al. [1]. This measure aims to reward the selection of relevant features while penalizing the inclusion of irrelevant ones. The success rate is calculated by the following formula:

$$Suc. = \left[\frac{R_s}{R_t} - \alpha\frac{I_s}{I_t}\right]$$

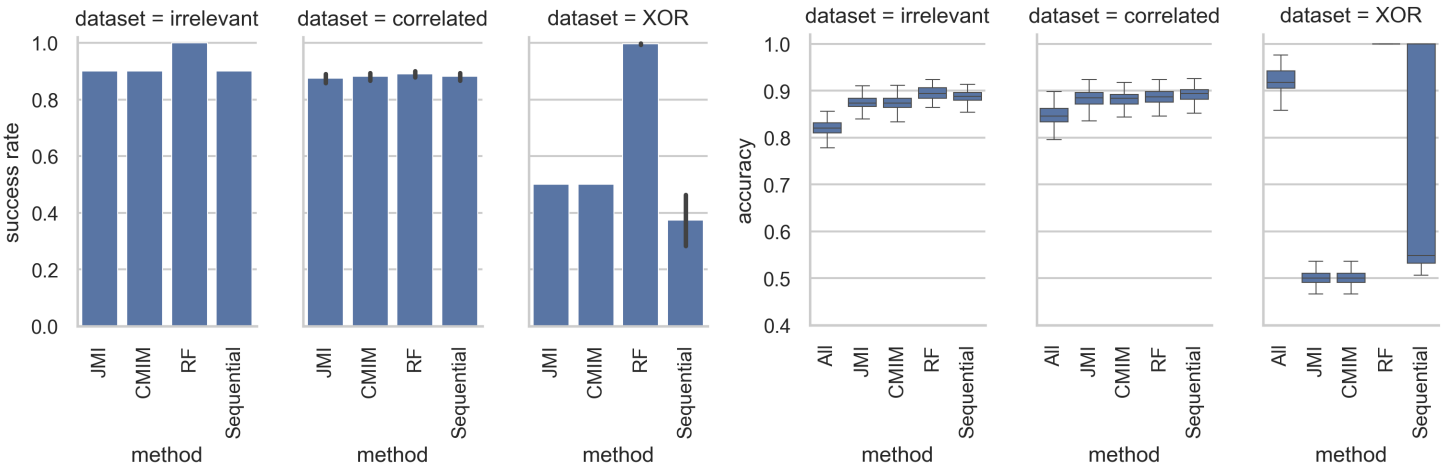
where R_s is the number of relevant features selected, R_t is the total number of relevant features, I_s is the number of irrelevant features selected, I_t is the total number of irrelevant features and $\alpha = \{\frac{1}{2}, \frac{R_t}{I_t}\}$.

The final assessment method concerns the stability of feature selection criteria. I employed the Kuncheva consistency index, as described in Brown et al. [2], which measures the consistency between two subsets.

3 Experiments & Results

This section is dedicated to experiments and results. I compare all feature selection methods on three artificial and three real-world datasets using various assessment techniques, as described in Section 2. Experiments using artificial datasets were repeated 100 times each, while experiments involving real-world data examples were repeated 50 times each.

Artificial datasets



(a) Success rate

(b) Accuracy

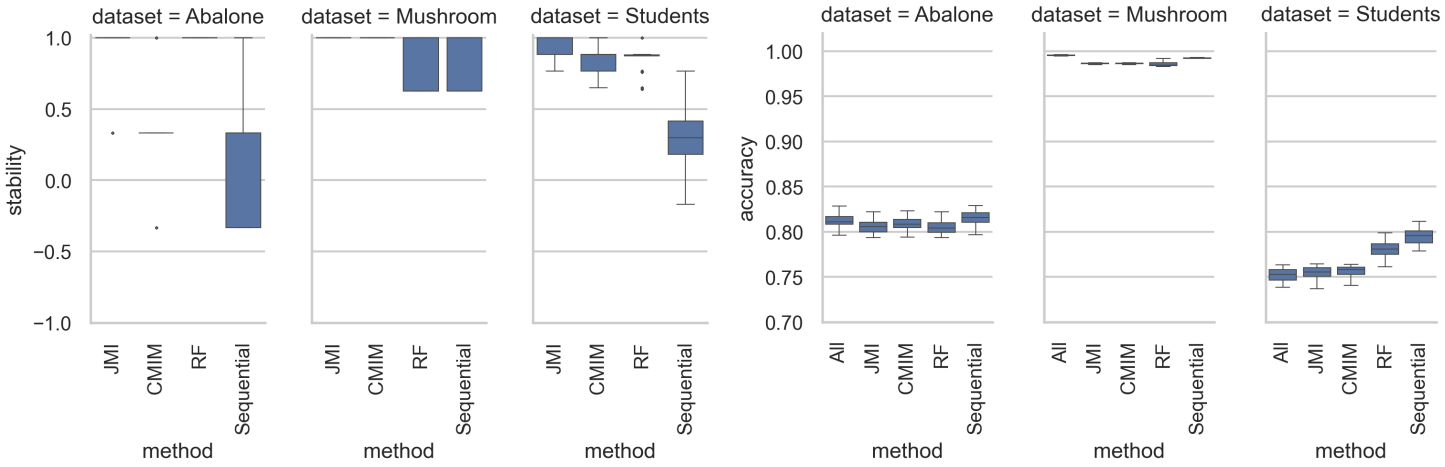
Figure 1: Success Rate and accuracy computed on artificial datasets.

In Figure 1a, the success rates achieved by feature selection methods on artificial datasets are presented. For the *irrelevant* dataset, all methods performed quite well, with Random Forest feature importance being the best. I checked that JMI and CMIM produced slightly worse results because these methods selected too many features. This indicates a need to reconsider my stopping rule. For the *correlated* dataset, all methods achieved similar results. Some variation occurred because highly correlated features were sometimes selected instead of the correct ones. In the last dataset, *XOR*, the results of the JMI and CMIM criteria were poor. Each time, they selected only one feature - X_0. As expected, mutual information criteria couldn't capture the more complex relationship between features. On the other

hand, Random Forest feature importance handled this problem almost perfectly. The behavior of the sequential method was more random for this dataset.

In Figure 1b, the accuracy achieved by a generic classifier (simple nearest neighbor) on the selected subset of features is visualized. For the *irrelevant* and *correlated* datasets, feature selection methods performed very well, achieving higher accuracy than the classifier built using all features in all cases.

Real-world data examples



(a) Stability

(b) Accuracy

Very little can be concluded from experiments performed on real-world data examples. I suspect that the selected datasets do not require much feature selection work. They do not contain an excessive number of features, and all features are probably more or less relevant, so reducing the set of variables generally does not improve the results. This is quite evident for the *Abalone* and *Mushroom* datasets, as shown in Figure 2b, which presents the accuracy achieved by a generic classifier—the accuracy on the full set of features is the highest. Some improvement can be seen for the third dataset, *Students*, where Random Forest feature importance and sequential methods significantly improved the results.

Even less can be said about methods' stability. For these three datasets, I obtained unclear results (Figure 2a). I can only say that JMI seems to be the most stable criterion, while the Sequential method is the least stable. To compare stability thoroughly, one should utilize a much larger number of datasets.

4 Conclusions

The project was performed according to the instructions. Feature selection methods were implemented, datasets were constructed or found, and experiments were conducted and assessed. Without a doubt, all feature selection methods were working, but to fully compare their performance, a much more extended study should be conducted.

References

[1] Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. A review of feature selection methods on synthetic data. *Knowledge and information systems*, 34:483–519, 2013.

[2] Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The journal of machine learning research*, 13(1):27–66, 2012.