Status report.
Feature selection package with IHT

Konstantin Pakulev, Viacheslav Pronin

April 2021

1 Project overview

Feature selection is of great importance for machine learning tasks especially for those that deal with high-dimensional data. There are many feature selection methods available, however, all of them are implemented in different libraries and lack a unified API. As all of the methods are different in their performance we want to select the best, however, the aforementioned problem makes it hard to perform a fair comparison of the existing methods.

Our project aims to mitigate the aforementioned issue by presenting a framework for evaluating different feature selection methods. We provide a baseline permutation importance [2] method in our pipeline (using implementation from sklearn) as well as implementations of two more recent methods: Normalised Iterative Hard Thresholding [1] and Simultaneous Feature and Feature Group Selection through Hard Thresholding [4]. The pipeline allows to perform evaluation on both synthetic and real data.

To setup, the structure of our framework as well as ensure reproducibility and the ease of maintenance of our project we use Kedro [3]. The implementation is available here: `https://github.com/adasegroup/FES-feature-selector`

2 Problem statement

The problem of feature selection can be formulated as follows. Let $A \in R^{n \times p}$ be a design matrix, $y \in R^n$ be a set of observations, $x \in R^p$ be a set of parameters and $\epsilon \in R^n$ be some noise. Assuming that observation are generated via $y = Ax + \epsilon$ we want to find such $\hat{x} \in R^p$ that satisfies optimization problem 1 where s_1 is some non-zero value.

$$\begin{aligned} &\underset{\hat{x}}{minimize} \quad \frac{1}{2}||A\hat{x} - y||_2^2 \\ &subject\ to \quad \sum_{j=1}^{p} I(|\hat{x}_j| \neq 0) \leq s_1 \end{aligned} \tag{1}$$

Sometimes it also can be beneficial to select feature groups along with features, i.e. when the data has some grouping structures. In such cases the optimization problem is complemented by an additional constraint 2 where the entities of x are separated into $|G|$ mutually exclusive groups, G_j denotes indices that belong to the j-th group and s_2 is some non-zero value.

$$\textit{subject to} \quad \sum_{j=1}^{|G|} I(||\hat{x}_{G_j}||_2 \neq 0) \leq s_2 \tag{2}$$

Solving the optimization 1 is a combinatorial problem for which no efficient algorithm exists for the general case. Instead sub-optimal algorithms are used that either convert the discrete constraints in 1 to their continuous surrogates or compute a local solution directly. In our project we will implement two methods that use the latter approach.

3 Project objectives

In section we introduce main challenges that we're going to face during the work on our project: methods implementations and evaluation protocol.

3.1 Baseline method

3.1.1 Permutation importance

Permutation importance is an approach to compute feature importances for any black-box estimator by measuring the decrease of a score when a feature is not available. Permutation importance is calculated after a model has been fitted (i.e., the estimator is required to be a fitted estimator compatible with scorer). This method contains the following steps:

- A baseline metric, defined by scoring, is evaluated on a data set defined by the X - the data set used to train the estimator or a hold-out set;
- A feature column from the validation set is permuted and the metric is evaluated again;
- The difference between the baseline metric and metric from permutation of the feature column is the permutation importance.

The mean of feature importance shows the degree of model performance accuracy deterioration with a random shuffling, and the standard deviation shows the variation of performance from one reshuffling to the next. Cases of negative values for permutation importances are possible, especially in small datasets; however, they merely depict the insufficiency of dataset, as they point out that "noisy" data happened to be more accurate than the real data, i.e. there is some random chance distortion.

This method is most appropriate for computing feature importances with a reasonably limited number of columns (features), as it can be resource-intensive.

3.2 Methods to-be-implemented

3.2.1 Normalized Iterative Hard Thresholding

The algorithm searches for $\hat{x}^{n+1}$ starting from $\hat{x}^0 = 0$ by applying the iterative procedure 3 where $H_K()$ is a non-linear operator that sets all but the top-k largest elements by their magnitude to zero and μ is computed adaptively as described in [1].

$$\hat{x}^{n+1} = H_K(\hat{x}^n + \mu A^\top (y - A\hat{x}^n)) \tag{3}$$

Pros:

- Gives theoretical guarantees on convergence to a local minimum of the cost function
- Doesn't depend on the scaling of the design matrix A

Cons:

- Theoretical guarantees on convergence exist only if A that satisfies restricted isometry property

3.2.2 Simultaneous Feature and Feature Group Selection through Hard Thresholding

The algorithm employs the iterative procedure 4 where f is the objective loss function, L is found by line search and $SGHT()$ stands for Sparse Group Hard Thresholding that is solved by dynamic programming as described in [4].

$$\hat{x}^{n+1} = SGHT(\hat{x}^n - \frac{1}{L}\nabla f(\hat{x}^n)) \tag{4}$$

Pros:

- Line search on each iteration can be significantly sped up
- Gives theoretical guarantees on convergence to a local minimum of the cost function that is at least within $c||y - Ax^*||_2$ radius from globally optimal solution x^* for a certain constant c

Cons:

- Theoretical guarantees on convergence exist only if A that satisfies restricted isometry property

3.3 Evaluation protocol

3.3.1 Metrics

To evaluate performance of different methods we plan to employ the protocol from [4]. Namely, we're going to report the number of selected features, feature groups (for ISTA with SGHT [4]) and mean squared error.

We also plan to study the influence of the noise on the performance of each method. Following [1] we aim to generate data with a certain signal-to-noise ratio (SNR) and compare the estimation of SNR to an oracle to which the noise values are known.

3.3.2 Synthetic data

As proposed in [4] for generating synthetic data for examination of methods we're going to use the linear model $y = Ax + \epsilon$, where the design matrix $A \in R^{100\times200}$ and the noise term ϵ follows normal distribution with ground truth x being partitioned into 20 equally sized groups. In this research, we intend to consider several kinds of grouping structures. The goal is to obtain an accurate (in terms of least squares) estimator of x that preserves the grouping structure, given only A and y.

3.3.3 Real data

Motivated by [4] we intend to study the algorithms on the Boston Housing data set. The original data set is used as a regression task, containing 506 samples with 13 features. Up to third-degree polynomial expansion is applied on each feature to account for the non-linear relationship between variables and response. For each variable x, x^2 and x^3 are recorded and gathered in a group. As a next step we split the data into the training set (approximately 50%) and testing set. The parameter settings for each method are properly scaled to fit the data set. We intend to use a linear regression model for training and testing with the evaluation protocol described in subsection 3.3.1.

4 Team members roles

In this research, we split responsibilities as follows:

- Viacheslav Pronin will implement the Normalized Iterative Hard Thresholding method and study the algorithms on the real data;
- Konstantin Pakulev will apply the Simultaneous Feature and Feature Group Selection through Hard Thresholding method and focus on the examination of algorithms on the synthetic data.

Both researchers will work on the metrics section and provide the evaluation of methods' accuracy, as well as conclusions and recommendations for further study.

References

[1] T. Blumensath and M. E. Davies. Normalized iterative hard thresholding: Guaranteed stability and performance. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):298–309, 2010.

[2] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[3] Lorena Bălan, Kiyohito Kunii (Kiyo), Dmitrii Deriabin, Lim Hoang, Andrii Ivaniuk, Yetunde Dada, Deepyaman Datta, Zain Patel, Gordon Wrigley, Ivan Danov, Jo Stichbury, Nasef Khan, Nikolaos Tsaousis, Merel Theisen, Waylon Walker, Tam Nguyen, Richard Westenra, Lais Carvalho, Marcelo Duarte Trevisani, Sebastian Bertoli, Shahil Mawjee, sasaki takeru, Bas Nijholt, Dmitry Vukolov, Kody Fischer, Vijaykumar, Yusuke Minami, bru5, and dr3s. quantumblacklabs/kedro: 0.17.0, December 2020.

[4] Shuo Xiang, Tao Yang, and Jieping Ye. Simultaneous feature and feature group selection through hard thresholding. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 532–541, New York, NY, USA, 2014. Association for Computing Machinery.