Supplemental materials to the paper 'Some proposal of the high dimensional PU learning classification procedure'

Konrad Furmańczyk, Marcin Dudziński and Diana Dziewa-Dawidczyk

Datasets

We consider the following real datasets for binary classification, with n denoting the number of instances and p standing for the number of attributes:

(D1) 'lymphona' dataset from 'spls' R package; $n = 62, p = 4026$; we merged 9 samples of follicular lymphoma and 11 samples of chronic lymphocytic leukemia into one 0-class; 42 samples of diffuse large B-cell lymphoma formed 1-class; matrix of attributes relates to the gene expression data;

(D2) 'prostate' dataset from 'spls' R package; $n = 102, p = 6033$; this dataset consists of 52 prostate tumor (1-class) and 50 normal samples (0-class); matrix of attributes relates to the gene expression data;

(D3) 'AlonDS' dataset from 'HiDimDA' R package; $n = 62, p = 2000$; this dataset consists of 2000 genes measured on 62 patients: 40 diagnosed with colon cancer (1-class) and 22 healthy patients (0-class);

(D4) 'dhfr' dataset from 'caret' R package; $n = 325, p = 228$; this data related to dihydrofolate reductase inhibition. This dataset contains values for 325 compounds (attributes). For each compound, 228 molecular descriptors (samples) have been calculated. Additionally, each sample is labeled as 'active' (0-class) or 'inactive' (1-class).

To create the PU datasets, we randomly selected $c \cdot 100\%$ of the labeled observations of S, for $c = 0.3; 0.5; 0.7; 0.9$. We split each dataset into the training set and the testing set, in the proportions 80% and 20%,

Table 1: Results for D1-D4

c	**detaset**	**scen 1**	**scen 2**	**svm**	**logit**
0.3	D1	0.051 (0.190)	0.158 (0.038)	0.848 (0.015)	0.843 (0.031)
0.5	D1	0.823 (0.121)	0.847 (0.008)	0.961 (0.040)	0.847 (0.008)
0.7	D1	0.892 (0.093)	0.998 (0.023)	0.998 (0.015)	0.924 (0.008)
0.9	D1	0.955 (0.056)	0.999 (0.011)	1 (0.000)	0.948 (0.036)
0.3	D2	0.401 (0.261)	0.524 (0.017)	0.285 (0.005)	0.711 (0.024)
0.5	D2	0.664 (0.148)	0.410 (0.074)	0.405 (0.119)	0.644 (0.071)
0.7	D2	0.769 (0.117)	0.762 (0.005)	0.857 (0.144)	0.785 (0.025)
0.9	D2	0.864 (0.100)	0.905 (0.000)	0.950 (0.019)	0.999 (0.005)
0.3	D3	0.578 (0.163)	0.385 (0.008)	0.384 (0.008)	0.692 (0.000)
0.5	D3	0.628 (0.128)	0.692 (0.083)	0.571 (0.060)	0.804 (0.048)
0.7	D3	0.655 (0.107)	0.538 (0.033)	0.459 (0.032)	0.845 (0.015)
0.9	D3	0.670 (0.107)	0.654 (0.039)	0.575 (0.043)	0.923 (0.000)
0.3	D4	0.653 (0.173)	0.608 (0.090)	0.484 (0.124)	0.529 (0.030)
0.5	D4	0.761 (0.107)	0.599 (0.018)	0.588 (0.020)	0.719 (0.021)
0.7	D4	0.850 (0.057)	0.792 (0.038)	0.822 (0.023)	0.868 (0.029)
0.9	D4	0.890 (0.038)	0.885 (0.023)	0.862 (0.015)	0.954 (0.015)

respectively. For all selected real datasets, the LassoJoint method was implemented. In the first step of our procedure, we used the Lasso method for the same scenarios as in the case of previously considered simulated synthetic datasets (see models (M1)-(M6) in the main text of our paper). For comparison, we applied AdaSampling scheme (see Yang et al. (2019)) together with support vector machine (SVM) and logistic regression.
Next, we determined the accuracy percentage and its standard deviations based on 100 MC replications of our experiments. The accuracy and standard deviation for the considered experiments on testing set are collected in Table 1.

Conclusions. The results of our experiments for real datasets show that if c increases, then the percentage of correct classifications increases as well. They also show that the all of the considered classification methods provide similar classification accuracy. In general, the LassoJoint method applied for scenario 2 works better than for scenario 1 and this method outperforms the rest of methods for D4 dataset. In turn, on the remaining data sets, the AdaSampling method used with the logistic regression or SVM performs better than our proposed method.