Can PCA extract important information from not significant features - Neural Network case

Project proposal

Maciej Odziemczyk

WNE UW

Introduction

Principal Component Analysis is a statistical method that produces a different view of the data. In case of many features there is a risk of high correlation, multicolinearity and others what causes a lot of noise in the data. If our algorithm is not robust to such factors, these noisy features may cause worse results. Tree-based algorithms tend to be quite robust, but gradient-based methods like logistic regression, Neural Networks etc. suffer from bad quality data. The main idea is to check if PCA on the "bad" features improves the Neural Network performance.

Dataset

Polish companies bankruptcy dataset is selected for this project and it is avalible to download from Machine Learning Repository, link. The reasons:

- Bankruptcy prediction problem is extremely important and pretty interesting for the author,
- dataset contains 64 continuous features (financial indicators) and binary dependent variable, bankruptcy or not,
- high class imbalance and lots of missing values make this dataset non-trivial,
- number of observations $\sim$ 6000 for one year forecast horizon.

Methods

Importance checking methods:

- Random Forest feature importance,
- Mutual information for categorical dependent variable,
- Statistical general to specyfic method based on logistic regression and Likelihood Ratio test,
- Lasso logistic regression,
- Spearman rank correlation coefficient (it has low power in binary dependent variable case, but it may help in specific situations)

Data Imputation method:

- Iterative algorithm based on Proximity Matrix from Random Forest implemented by the author

Model comparison methods:

- Stratified 5-fold Cross Validation (train, test),
- average PR-AUC, ROC-AUC metrics for test set in CV

Models

The main models to compare:

- Extreme Gradient Boosting on all the data - very powerful and robust algorithm in many problems,
- Standard 2-3 layer Feedforward Neural Network on all the data - potentially non robust for noisy data,
- Standard 2-3 layer Feedforward Neural Network on PCA transformed data

auxiliary models:

- Random Forest - for data imputation, feature importance,
- Logistic Regression - for feature importance,
- Lasso Logistic Regression - for feature importance.

Hyperparameters optimization

Hyperparameters optimization procedure:

- based on the Cross Validation - train, test set (5 folds)
- sequential check of a wide range of values,
- random search on a narrow range.

Hyperparameters to tune:

- XGB:
  - max trees depth,
  - subsample,
  - column sample by tree,
  - column sample by level,
  - regularization parameter λ,
  - γ parameter,
  - η parameter,
  - rate drop,
  - skip drop
- Neural Networks:
  - number of hidden layers,
  - number of hiden units,
  - activation function,
  - regularization,
  - dropout,
  - batch size,
  - optimizer

Packages

- Pandas,
- Numpy,
- Scipy,
- Matplotlib,
- Seaborn,
- Sklearn,
- Statsmodels,
- Pickle,
- Time,
- Random,
- TensorFlow,
- Keras,
- Xgboost.

Inspirations

- Maciej Zieba, Sebastian K. Tomczak, Jakub M. Tomczak. Ensemble Boosted Trees with Synthetic Features Generation in Application to Bankruptcy Prediction. *Expert Systems With Applications*, 58(1): 93–101, 2016
- Mu-Yen Chen. Predicting corporate financial distress based on integration of decision tree classification and logistic regression. *Expert Systems With Applications*, 38: 11261–11272, 2011