Outlier Detection Toolbox

Göker Erdoğan

May 22, 2012

OUTLIER DETECTION TOOLBOX IN MATLAB

For the evaluation of our spectral outlier detection algorithm, we have developed an outlier detection toolbox, `odToolbox`[1], in MATLAB[2]. It features:

- Implementations of the outlier detection methods; Active-Outlier [1], Local outlier factor [2], Parzen windows [3], Feature Bagging [4] and decision tree[3]
- Implementations of the spectral methods; Principal components analysis [5], Laplacian eigenmaps [6], Multidimensional scaling [7] and Kernel principal components analysis [8]
- A data set format, routines to read and pre-process data sets
- An experiment result format and functions for calculation of AUC for ROC and precision-recall (PR) curves
- Routines to visualize discriminants of methods, plot ROC and PR curves [4]

The source code is properly documented and information on any function can be seen by calling `help functionName`. We provide two GUIs for demonstration under `demo` folder. First demonstration, `demo.m`, lets user to choose points in 2D, input kernel parameters and plots the spectral transformations. This demonstration can be started by typing `demo` in the MATLAB command line. A sample run of this demonstration can be seen in Figure 1 and 2.

In the second demonstration, `od.m`, the user is able to run outlier detection methods, AO, LOF and PW with PCA, LEM, MDS and KPCA (without mean centering) on a chosen data set. The user needs to select a folder containing the necessary data set files. Each data set requires five files, these are:

- `tvx.mat`: an $N \times d$ MATLAB matrix containing training and validation instances. Instances are on the rows and attributes are in the columns.

[1] `odToolbox` can be downloaded from http://gokererdogan.com/files/thesis/odtoolbox.zip

[2] http://www.mathworks.com

[3] Decision tree is implemented by `classregtree` class which is available in MATLAB `statistics` toolbox

[4] ROC curves are calculated with `croc` and `auroc` functions implemented by Dr Gavin C. Cawley taken from http://theoval.cmp.uea.ac.uk/matlab/

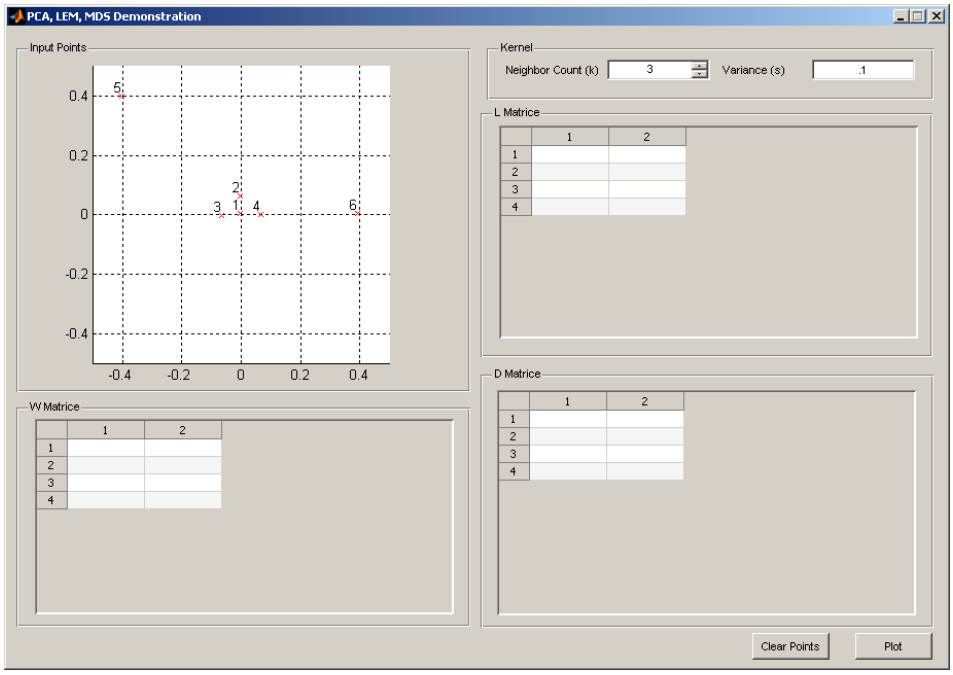

Figure 1: Spectral methods demonstration `demo` script's first screen where points and kernel parameters are selected.

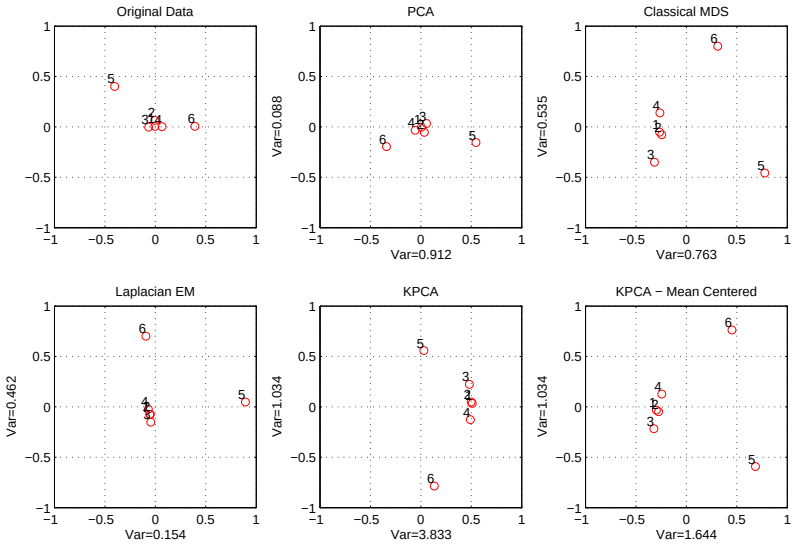

Figure 2: Spectral methods demonstration `demo` script's second screen where spectral transformations are plotted.

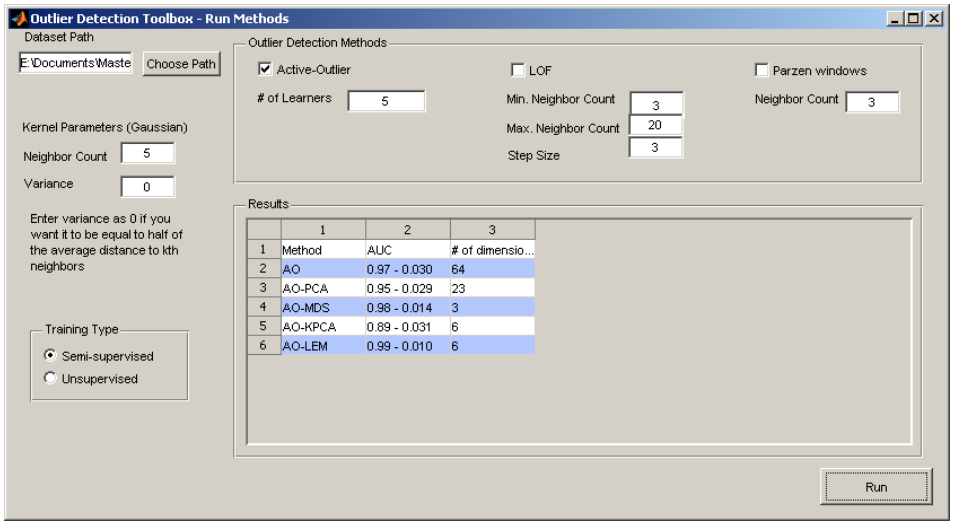

Figure 3: Outlier detection toolbox demonstration `od` script's GUI.

- `tvy.mat`: an N-vector of class labels for training/validation set
- `tsx.mat`: an $N_{test} \times d$ MATLAB matrix containing test instances
- `tsy.mat`: an N_{test}-vector of class labels for test set
- `def.txt`: data set definition file where first line contains the data set name, second line class labels separated by space for each typical class, third line includes class labels for outlier class and fourth line lists the indices of categorical attributes. A sample file can be seen under `datasets/optdigits1` folder

The parameters for the Gaussian kernel, semi-supervised or unsupervised training selection and method specific parameters are also input by the user as seen in Figure 3. The selected methods are evaluated on the chosen data set with no transformations, with PCA, LEM, MDS and KPCA. For spectral methods, the number of dimensions that give the best performance is found by cross validation and the AUC on the test set is calculated with this dimensionality. The results are given in the GUI after the experiments finish. Additionally, the MATLAB structures containing the results for each run and ROC, PR curve plots are saved into the data set folder. A corresponding script to `od.m`, `od_script.m`, is also provided that shows how to run experiments with different methods and parameters. `odToolbox` is distributed under the GNU General Public License[5], it can be redistributed and modified freely for non-commercial use. We gently request users of this toolbox to reference this work in the resulting publications.

[5] http://www.gnu.org/licenses/

References

[1] Abe, N., B. Zadrozny and J. Langford, "Outlier Detection by Active Learning", *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pp. 504–509, ACM, New York, NY, USA, 2006.

[2] Breunig, M. M., H. P. Kriegel, R. T. Ng and J. Sander, "LOF: Identifying Density-based Local Outliers", *SIGMOD Record*, Vol. 29, pp. 93–104, May 2000.

[3] Parzen, E., "On Estimation of a Probability Density Function and Mode", *The Annals of Mathematical Statistics*, Vol. 33, No. 3, pp. pp. 1065–1076, 1962.

[4] Lazarevic, A. and V. Kumar, "Feature Bagging for Outlier Detection", *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pp. 157–166, ACM, New York, NY, USA, 2005.

[5] Jolliffe, I., *Principal Component Analysis*, Springer Series in Statistics, Springer-Verlag, 2002.

[6] Belkin, M. and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation", *Neural Computation*, Vol. 15, No. 6, pp. 1373–1396, 2003.

[7] Cox, T. and M. Cox, *Multidimensional Scaling*, Monographs on Statistics and Applied Probability, Chapman & Hall, 1994.

[8] Schölkopf, B., A. J. Smola and K.-R. Müller, "Kernel Principal Component Analysis", *ICANN*, pp. 583–588, 1997.