COSC 425/528 — Introduction to Machine Learning — Fall 2018

Project 5: Support Vector Classification

Due Dec. 8, 2017

In this project you will apply support vector classification (SVC) to three classification problems. The files are in the `Project 5` folder; the attributes are described in the `.name` files, also in the folder. You will not need to program your own SVC, because you will use off-the-shelf library functions. In this way you will get some experience using available machine learning resources.

- For python, use `SVC` in `scikit-learn`. For a general introduction see http://scikit-learn.org/stable/tutorial/basic/tutorial.html. For documentation on `SVC`, see http://scikit-learn.org/stable/modules/svm.html. The `C` parameter is the error cost (or penalty factor) and the `gamma` parameter controls the scaling of the kernel function. For standardizing data you can use `StandardScaler` (http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html). For grid searches, use `GridSearchCV` (http://scikit-learn.org/stable/modules/grid_search.html).

- For Matlab, use `fitcsvm` (for binary classification) or `fitcecoc` (for multiple classes). For a general introduction to SVMs in Matlab, see https://www.mathworks.com/help/stats/support-vector-machine-classification.html. This page links to documentation for `fitcsvm` and `fitcecoc`. The `BoxConstraint` parameter is the error cost (or penalty factor) and the `KernelScale` parameter controls the scaling of the kernel function. For standardization you can use the `Standardize` option on these functions or use the `zscore` function. You can program your own grid search, use the `OptimizeHyperparameters` option on these functions, or use `svm_grid_search` (https://www.mathworks.com/matlabcentral/fileexchange/50869-svm-grid-search).

- For other languages, you can use `LIBSVM`, but you will have to install it. See https://www.csie.ntu.edu.tw/~cjlin/libsvm/ and scroll down to the bottom of the page. If you go this route, I advise starting early, since there can be complications installing packages such as this. You can use `grid.py`, which is in the `Tools` directory of `LIBSVM`, for grid searches.

In this project you will apply SVC to three different datasets.

(1) SVM works best on standardized data. You can standardize attributes to [0, 1], [–1, 1], or $(\mu=0,\sigma=1)$. As usual, calculate your standardization parameters on the training data and use them also on the validation and testing data.

(2) Your first task is a binary classification problem: predicting “good” vs. “bad” interactions of radar signals with electrons in the ionosphere. The last attribute is the label (“`g`” or “`b`”) and the other 34 are measurements. There are 351 instances in `ionosphere.data` in the `Project 5` folder. You will have to split them into training, validation, and testing subsets. For more information, see https://archive.ics.uci.edu/ml/datasets/ionosphere.

(3) Perform a *coarse grid search* to find the best range of hyperparameters (specifically, penalty and kernel scale), and then a *fine grid search* to find the optimal hyperparameters for this problem. Report your results and classification performance for these hyperparameters.

(4) The second task is a multiclass problem, with 11 classes (vowel sounds) and 10 features (https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+%28Vowel+Recognition+-+Deterding+Data%29). In the file `vowel-context.data`, the first three attributes are irrelevant and the last is the vowel label. As in step (3), split the file and perform grid searches to optimize your hyperparameters.

(5) The third task is also multiclass, with 7 classes and 36 features: determine the type of terrain from multispectral values of pixels in 3×3 neighborhoods in satellite images [see https://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite)]. There are 4435 training instances in `sat.trn` and 2000 testing (validation) instances in `sat.tst`. In the files the last attribute is the label (1–7, but this sample contains no instances of class 6). As in step (3), perform grid searches to optimize your hyperparameters.