Homework VI

Due date: Nov. 2, 2021 (11:59pm)

Software

To train a regularized logistic regression and support vector machine, you can use the liblinear library `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`. To train a kernel SVM, you can use the libsvm library `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. Many interfaces are available including Matlab, Python, Octav, Java, R, etc. Read the documentation that is provided in the webpage or the package (in particular the README file in the folder for your used interface). To compute AUC measure and plot the ROC curve, you can use Matlab function "perfcurve" or Python module sklearn.metrics. **Please report what interface you use.**

Problem 1: Regularized Logistic Regression (35 points)

Let $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be the training examples, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$. The negative log-likelihood of the regularized logistic regression, denoted by $L(\mathbf{w})$ is written as

$$L(\mathbf{w}) = C \sum_{i=1}^{n} \ln(1 + \exp(-y_i \mathbf{x}_i^\top \mathbf{w})) + \frac{1}{2}\|\mathbf{w}\|_2^2$$

where C is a parameter that controls the balance between the loss and the regularization. The optimal solution for $\mathbf{w} \in \mathbb{R}^d$ is obtained by minimizing $L(\mathbf{w})$.

- Show $w_k = w_l$ for the optimal solution $\mathbf{w}$ if two attributes k and l are identical, i.e., $x_{i,k} = x_{i,l}$ for any training example $\mathbf{x}_i$.
- Train and test a regularized logistic regression model one two data sets, namely the breast cancer and sonar data sets which are available here `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#breast-cancer` and here`https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#sonar`. We will use the scaled version for our experiment. A copy of them is also enclosed in this homework. Use the provided training/testing splitting. In particular, the file "xxx-scale-test-indices.txt" contains the indices of examples in the original file for training, and "xxx-scale-test-indices.txt" contains the indices of examples in the original file for testing.
    - Use the 5-fold cross validation method to decide the best value of the parameter C. The candidate values for C are $0.1, 1, 10, 100, 1000$. For each C, report the training error and validation error. Choose the best C that yields the lowest validation error.
    - Use the selected best C value to train a logistic regression model on the whole training data and evaluate and report its performance (by error rate) on the testing data.

  - Report the results on the two data sets.

Note: To train a regularized logistic regression by liblinear library, you can use the option "-s 0".

Problem 2: Support Vector Machine (35 points)

- Repeat the same experiments as in Problem 1 by using linear SVM. To train a linear SVM by liblinear you can use the option "-s 3".
- Repeat the same experiments as in Problem 1 by using kernel SVM. To train a kernel SVM by libsvm you can use the option "-s 0". Use the optional "-t " to choose different types of kernels. Try polynomial kernel and RBF kernel with default values of parameters.
- Compare the test error given by Logistic Regression, Linear SVM and Kernel SVMs.

Problem 3: Data Preprocessing (30 points)

You are going to use the covtype data set in this question and the next question. This data is described here `https://archive.ics.uci.edu/ml/datasets/Covertype`. The raw data ("covtype.data") is provided in the data folder and a training/testing splitting is also provided (see covtype.train.index and covtype.test.index). Each row in the data file consists of 54 features (the first 54 columns) and the label (the last column). The original data is for a multi-class classification. There are a total of 6 classes. For this problem, you will build a classifier for classifying label "2" (positive) vs the rest (negative).

In order to calculate AUC, you need prob(y) in a class. This is only supported for logistic regression in liblinear

You are asked to compare different data preprocessing methods. There are three commonly used data preprocessing methods, rescaling, standardization and normalization. For more details, please read here `https://en.wikipedia.org/wiki/Feature_scaling`. If we let $X \in \mathbb{R}^{n\times d}$ denote the data matrix (n examples and d features), note that the first two methods for conduced for each column and the normalization is conducted for each row. You need to use the same code from the last problem to train a linear SVM classifier ("-s 3" in liblinear) on the training data with the 5-fold cross-validation to find the best C.

- Submit your code for data preprocessing, training, and evaluation (mark you code clearly for each functionality).
- Report in a table the accuracy, F1-score, AUC on the testing data for using each preprocessed data and the raw data.
- Plot in a figure the ROC curves for using each preprocessed data and the raw data.
- Discuss your observations of the results.

Problem 4: Dual Form and Kernel Trick (20 points, optional)

In Homework II, we derive a regression model using the Laplacian noise model. It is cast into the following optimization problem (also known as regularized least absolute deviation):

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \frac{1}{n}\sum_{i=1}^{n} |y_i - \mathbf{w}^\top \mathbf{x}_i|$$

Derive a dual formulation of the above problem that is quadratic in terms of dual variables $\alpha \in \mathbb{R}^n$, and then apply the kernel trick to derive the kernelized least absolute deviation. (Hint: use $|s| = \max_{\alpha\in[-1,1]} \alpha s$ to tackle the loss part).