CS512 Machine Learning, Fall 2019: Homework 2

Due: Dec 8 22:00 pm

Instructions

- Submit an online copy of your report and code through SUCourse.
- Please submit the report and the code separately. Name your submission as CS512-YourName.pdf and CS512-YourName-Code.zip (or .tar), where you substitute in your first and last names into the filenames in place of 'YourName'.
- You may code in any programming language you would prefer. For this homework, you are allowed to use libraries.
- If you are submitting the homework late (see the late submission policy in syllabus), we will grade your homework based on the time stamp of submission and your remaining late days.

1 Ensembles [15 pts]

Consider that you have trained M different linear regressors, each one denoted by $f_i(x) : R^d \rightarrow R$ where $i \in 1...M$. For a given test example x with target groundtruth value y, the error ϵ_i of each regressor $f_i(x)$ is measured by the squared error:

$$\epsilon_i(x) = (f_i(x) - y)^2 \tag{1.1}$$

The average error is then simply given by the following formula:

$$\epsilon_{\text{AVG}}(x) = \frac{1}{M}\sum_{i=1}^{M} \epsilon_i(x) \tag{1.2}$$

You design a committee regressor $F(x)$ that is defined as the average of individual regressors:

$$F(x) = \frac{1}{M}\sum_{i=1}^{M} f_i(x) \tag{1.3}$$

Similarly, the error for the committee regressor is given by the following formula:

$$\epsilon_{\text{COM}}(x) = (F(x) - y)^2 \tag{1.4}$$

Prove that the error of the committee regressor is always below the average error of the individual regressors, that is:

$$\epsilon_{\text{COM}}(x) \leq \epsilon_{\text{AVG}}(x) \tag{1.5}$$

Hint: You can use the following simplified case of the Jensen's inequality in your proof: for a given convex function g and a set of numbers $\{z_1, ..., z_n\}$, the Jensen's inequality says that

$$g\left(\frac{1}{n}\sum_i z_i\right) \leq \frac{1}{n}\sum_i g(z_i) \tag{1.6}$$

always holds true.

2 Kernels [15 pts]

The inner product between two infinite vectors $a = \langle a_1, a_2, \dots \rangle$ and $b = \langle b_1, b_2, \dots \rangle$ is defined as the infinite sum given in 2.1 (assuming the series converges).

$$K(a, b) = a \cdot b = \sum_{k=0}^{\infty} a_k b_k \tag{2.1}$$

Can we explicity compute $K(a, b)$? What is the explicit form of $K(a, b)$? *Hint: you may want to use the Taylor series expansion of e^x which is given in Equation 2.2.*

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} \tag{2.2}$$

3 Letter recognition using SVM [70 pts]

In this part, you will separate handwritten B characters from P characters in UCI letter recognition dataset[1], using Support Vector Machines (SVM). Download the zip file `HW2data` on Moodle and you will find `Bs.csv` and `Ps.csv`. Now, randomly divide the data for Bs and Ps into train and test (roughly 80% - 20% for each class, respectively)

You may use any SVM package you prefer (sckitlearn, LibSVM [2], Matlab or any other SVM package)

a) Linear Kernel [25 pts] Train an SVM classifier with linear kernel on your training set. You also need to fine-tune your classifier with cross-validation to find the best cost parameter (C). To do that, start from some small C value like 0.001 and step through larger values and perform cross-validation to measure the quality of each C value (Hint: You can try values like $10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}$). Plot cross-validation accuracy values over your tuning process. Report the highest cross validation accuracy, and the corresponding C value. Re-train a model using the best C value and run it on the test set. Output the decision values of test set as a file. Declare your test result.

b) Radial Basis Function Kernel [45 pts] Use SVM with RBF kernel. RBF kernel is

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2}\right) \tag{3.1}$$

$||\mathbf{x} - \mathbf{x}'||^2$ is the squared Euclidean distance between the two feature vectors. σ is a free parameter. An equivalent, but simpler, definition involves a parameter $\gamma = -\frac{1}{2\sigma^2}$:

$$K(\mathbf{x}, \mathbf{x}') = \exp(\gamma||\mathbf{x} - \mathbf{x}'||^2) \tag{3.2}$$

Kernel SVM requires C and γ parameters to be tuned ($\gamma = \frac{-1}{2\sigma^2}$). To do that, keep a C value constant as you are trying some range of γ values. Update C again and apply same procedure to γ and iterate up to some end condition. You may use each C and γ pair, where $C \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$, and $\gamma \in \{2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^{0}\}$. Perform cross-validation for each pair of parameters Plot cross validation accuracy values as a surface plot. Give the best cross-validation accuracy and the corresponding C and γ values.

Re-train the model on the full training set using the best parameter combination, and test the resulting model on test set. Indicate the test-set accuracy you obtain and compare your result with the Linear SVM result. Output the decision values of test set as a file. Is Kernel SVM better, why?

Add the plots to your report, label the axes and add titles to plots, add a legend if necessary.

[1] https://archive.ics.uci.edu/ml/datasets/Letter+Recognition
[2] http://www.csie.ntu.edu.tw/ cjlin/libsvm/