COMS E6998 section 012
PRACT DEEP LEARNING SYS PERF
Instructor: Parijat Dube
Due: October 2, 2021

Homework 1

Problem 1 - *Linear Separability* 10 points

Consider a dataset with two features x_1 and x_2 in which the points $(-1,-1),(1,1),(-3,-3),(4,4)$ belong to one class and $(-1,1),(1,-1),(-5,2),(4,-8)$ belong to the other.

1. Is this dataset linearly separable ? Can a linear classifier be trained using features x_1 and x_2 to classify this data set ? You can plot the dataset points and argue. (2)
2. Can you define a new 1-dimensional representation z in terms of x_1 and x_2 such that the dataset is linearly separable in terms of 1-dimensional representation corresponding to z ? (4)
3. What does the separating hyperplane looks like ? (2)
4. Explain the importance of nonlinear transformations in classification problems. (2)

Problem 2 - *Bias Variance Tradeoff, Regularization* 40 points

1. Derive the bias-variance decomposition for a regression problem, i.e., prove that the expected mean squared error of a regression problem can be written as

$$E[MSE] = Bias^2 + Variance + Noise$$

Hint: Let $y(x) = f(x) + \epsilon$ be the true (unknown) relationship and $\hat{y} = g(x)$ be the model predicted value of y. Then MSE over test instance x_i, $i = 1, \ldots, t$, is given by:

$$MSE = \frac{1}{t}\sum_{i=1}^{t}(f(x_i) + \epsilon - g(x_i))^2 \text{ (5)}$$

2. Consider the case when $y(x) = x + \sin(1.5x) + \mathcal{N}(0, 0.3)$, where $\mathcal{N}(0, 0.3)$ is normal distribution with mean 0 and variance 0.3. Here $f(x) = x + \sin(1.5x)$ and $\epsilon = \mathcal{N}(0, 0.3)$. Create a dataset of size 20 points by randomly generating samples from y. Display the dataset and $f(x)$. Use scatter plot for y and smooth line plot for $f(x)$. (5)
3. Use weighted sum of polynomials as an estimator function for $f(x)$, in particular, let the form of estimator function be:

$$g_n(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_n x^n$$

Consider three candidate estimators, g_1, g_3, and g_{10}. Estimate the coefficients of each of the three estimators using the sampled dataset and plot $y(x), f(x), g_1(x), g_3(x), g_{10}(x)$. Which estimator is underfitting ? Which one is overfitting ? (10)

4. Generate 100 datasets (each of size 50) by randomly sampling from y. Partition each dataset into training and test set (80/20 split). Next fit the estimators of varying complexity, i.e., $g_1, g_2, ...g_{15}$ using the training set for each dataset. Then calculate and display the squared bias, variance, and error on testing set for each of the estimators showing the tradeoff between bias and variance with model complexity. Can you identify the best model ? (10)
5. One way to increase model bias is by using regularization. Lets take the order 10 polynomial and apply $\mathcal{L}_2$ regularization. Compare the bias, variance, and MSE of the regularized model with the unregularized order 10 polynomial model ? Does the regularized model have a higher or lower bias ? What about MSE ? Explain. (10)

Homework 1

Note: For part 2 and 3 of this problem limit the range of x range for the 20 points generated to lie between some range, say 0 and 10, to observe overfitting and underfitting. Remember to use the same range for training and testing. Additionally, please note to sort the points (increasing x) before plotting. The graph must contain a scatter plot of the points and line plot of the functions.

For part 4 of this problem there are two different ways to sample x and y when creating 100 datasets.

- Follow the post https://dustinstansbury.github.io/theclevermachine/bias-variance-tradeoff. The idea is to keep the value of x same across all the 100 datasets. The y values will vary since it contains the noise (Normal distribution) component.
- Sample a test set (of size 10) before sampling any training dataset. Then sample training set (of size 40) for each 100 dataset but make sure that none of the 10 test set samples should show in any of the 100 datasets. So all the datasets share this common test set but their train set is different.

  The key is to have a fixed test set even though you have 100 independently sampled training set

Problem 3 - *OpenML, Algorithmic Performance Scaling* 25 points

OpenML (https://www.openml.org) has thousands of datasets for classification tasks. Select any 2 datasets from OpenML with different number of output classes.

1. Summarize the attributes of each dataset: number of features, number of instances, number of classes, number of numerical features, number of categorical features. (5)
2. For each dataset, select 80% of data as training set and remaining 20% as test set. Generate 10 different subsets of the training set by randomly subsampling $10\%, 20\%, \ldots, 100\%$ of the training set. Use each of these subsets to train two different classifiers: *Random forest* and *Gradient boosting*. When training a classifier also measure the wall clock time to train. After each training, evaluate the accuracy of trained models on the test set. Report model accuracy and training time for each of the 10 subsets of the training set. Generate learning curve for each classifier. A learning curve shows how the accuracy changes with increasing size of training data. Also create a curve showing the training time of classifiers with increasing size of training data. So, for each dataset you will have two figures: First figure showing learning curves (x-axis being training data size and y-axis accuracy) for the two classifiers and second Figure showing training time for the two classifiers as a function of training data size. (15)
3. Study the scaling of training time and accuracy of classifiers with training data size using the two figures generated in part 2 of the question. Compare the performance of classifiers in terms of training time and accuracy and write 3 main observations. Which gives better accuracy ? Which has shorter training time ? (5)

Problem 4 - *Precision, Recall, ROC* 25 points

This question is based on two papers, one from ICML 2006 and other from NIPS 2015 (details below). ICML paper talks about the relationship between ROC and Precision-Recall (PR) curves and shows a one-to-one correspondence between them. NIPS paper introduces Precision-Recall-Gain (PRG) curves. You need to refer to the two papers to answer the following questions.

1. Does true negative matter for both ROC and PR curve ? Argue why each point on ROC curve corresponds to a unique point on PR curve ? (5)

2. Select one OpenML dataset with 2 output classes. Use two binary classifiers (Adaboost and Logistic regression) and create ROC and PR curves for each of them. You will have two figures: one containing two ROC and other containing two PR curves. Show the point where an *all positive classifier* lies in the ROC and PR curves. An all positive classifier classifies all the samples as positive. (10)

3. NIPS paper defined PR Gain curve. Calculate AUROC (Area under ROC), AUPR (Area under PR), and AUPRG (Area under PRG) for two classifiers and compare. Do you agree with the conclusion of NIPS paper that practitioners should use PR gain curves rather than PR curves. (10)

Related papers:

- Jesse Davis, Mark Goadrich, The Relationship Between Precision-Recall and ROC Curves, ICML 2006.
- Peter A. Flach and Meelis Kull, Precision-Recall-Gain Curves: PR Analysis Done Right, NIPS 2015.