EE 660 **Midterm Assignment** Posted: Wed., 10/6/2021
Jenkins v2.0 Due: Wed., 10/13/2021, 11:59 PM PDT

Ground rules

For this assignment, you may use any of our EE 660 course materials, and you may use your computer for coding and accessing the assignemnt, EE 660 materials, piazza, and D2L. You may also use the internet for information about python coding including python libraries for this Midterm Assignment.

You are not allowed to use the internet to look up answers or partial answers to any of this assignment's problems, or communicate with other people (e.g., other students, people on discussion forums, etc.) on topics related to this assignment.

For all coding in this assignment, you are allowed to use built-in python functions, numpy, matplotlib, sklearn (and pandas only for reading/writing csv files).

For questions on this assignment, when posting on piazza, please consider whether this is a question that is appropriate for all students (e.g., clarifying the problem statement, what is allowed, suspected typos or omissions in the problem statement), or only for the instructors (e.g., something that includes your approach or solution) which should be posted as a "private" post that is only visibile to the professor, TAs and you.

Please respect your classmates and follow all of these guidelines, so that everyone can be graded fairly. Any violations that are detected will result in penalties.

Uploading your solutions. Please upload the following files before the final due date and time:

1. A single pdf file of all your soltions/answers.
2. A single computer-readable pdf file of all your code.

Problems and points. This Midterm Assignment has 3 problems, worth a total of 100 points possible. There will be partial credit on most problem parts. Good luck!

1. ***Generalization bounds: theory and experiment***

[created by Fernando Valladares Monteiro and Keith Jenkins]

You are given a dataset D_{full} with 10 features and 10^6 samples. We want to use a simple perceptron to test how the generalization bound, ε varies in practice as a function of the number of samples N_{in}, the tolerance level δ, and the VC dimension d_{vc}. Whenever you vary one of these parameters, the others should be kept constant.

Hint: Remember that the VC dimension of the perceptron equals the number of features plus one, i.e., $d_{vc} = d + 1$.

a) To study the dependence on the tolerance level use $N_{in} = 1000$, $d_{vc} = 4$, $\delta = 0.01, 0.02, \ldots 0.5$ and follows the steps:

i) Use the parameters above to compute theoretical generalization bound values as a function of δ.

ii) Extract a dataset D_{out} with $N_{out} = 10^5$ samples and the first d (note that $d \neq d_{vc}$) features from D_{full}. You will use this dataset to estimate E_{out}.

iii) Extract a dataset D_{in} with N_{in} samples and the first d features from D_{full}, train a perceptron on D_{in}, and compute E_{in} and E_{out}

iv) Repeat steps (iii) **1000** times, remembering to save all your results in arrays.

v) For each $\delta \in \{0.01, 0.02, \ldots 0.5\}$ in the values above, determine the maximum value of $|E_{out} - E_{in}|$ such that, over the 1000 runs, a proportion δ of the samples are above that value. Hint: for $\delta = 0.5$, this value is the median. This provides an estimate of the generalization bound as a function of the tolerance level.

vi) Plot the results from (i) and from (v) side by side.

Would you say the theoretical bound is tight, moderate, or loose in this experiment? (Let's define tight as within a factor of 2; moderate as a factor of 2-10; and loose as a factor of more than 10.) Would you say the relationship between the bound and the tolerance level is the same in theory and in the experiment, i.e., do the plots have similar shapes? If not, conjecture why if you can.

b) To study the dependence on VC dimension, you will vary the perceptron's complexity by selecting only the **first** d features. Use: $N_{in} = 1000$, $d_{vc} = [2 \ldots 11]$, $\delta = 0.1$ and follow the steps:

i) Use the parameters above to compute theoretical generalization bound values as a function of d_{vc}

ii) Extract a dataset $D_{out}^{(10)}$ with $N_{out} = 10^5$ samples and all the 10 features from D_{full}. You will use this dataset to estimate E_{out}.

iii) Extract a dataset $D_{in}^{(10)}$ with $N_{in} = 1000$ and all the features from the D_{full}.

iv) For each $d_{vc} = [2 \dots 11]$: create datasets $D_{out}^{(d)}$ and $D_{in}^{(d)}$ by extracting the first d (note that $d \neq d_{vc}$) features of $D_{out}^{(10)}$ and $D_{in}^{(10)}$ respectively, train a perceptron on $D_{in}^{(d)}$, and compute E_{in} and E_{out}.

v) Repeat steps (iii)-(iv) 100 times, remembering to save all your results in arrays.

vi) Determine the maximum value of $|E_{out} - E_{in}|$ such that, over the 100 runs, a proportion δ of the samples are above that value. This provides an estimate of the generalization bound as a function of the VC dimension.

vii) Plot the results from (i) and from (vi) side by side

Would you say the theoretical bound is tight, moderate, or loose in this experiment? (Let's define tight as within a factor of 2; moderate as a factor of 2-10; and loose as a factor of more than 10.) Would you say the relationship between the bound and the VC dimension is the same in theory and in the experiment, i.e., do the plots have similar shapes? If not, conjecture why if you can. (Hint: the model that the data was generated from is very simple.)

c) To study the dependence on the number of samples use: $N_{train} = [10,30,100,300,1000,3000,10000]$, $N_{test} = N_{train}/5$, $d_{vc} = 4$, $\delta = 0.1$ and follow the steps:

i) Use the parameters above to compute theoretical generalization bound values as a function of N_{train}.

ii) Use the parameters above to compute theoretical generalization bound values as a function of N_{test}. Remember that the generalization bound for train and test sets have different formulas.

iii) Extract a dataset D_{out} with $N_{out} = 10^5$ samples and the first d (note that $d \neq d_{vc}$) features fromD_{full}. You will use this dataset to estimate E_{out}

iv) For each N_{train}: extract a dataset D_{train} with N_{train} samples and d features from D_{full}, extract a dataset D_{test} with $N_{train}/5$ samples and d features from D_{full}, train a perceptron on D_{train}, and compute E_{train}, E_{test} and E_{out}.

v) Repeat step (iv) 100 times, remembering to save all your results in arrays.

vi) Determine the maximum value of $|E_{out} - E_{train}|$ such that, over the 100 runs, a proportion δ of the samples are above that value. This provides an estimate of the generalization bound as a function of the number of training samples.

vii) Repeat item (vi) for $|E_{out} - E_{test}|$.

viii) Plot the results from (i) and from (vi) side by side.

ix) Plot the results from (ii) and from (vii) side by side.

Would you say the theoretical bound is tight, moderate, or loose in this experiment? (Let's define tight as within a factor of 2; moderate as a factor of 2-10; and loose as a factor of more than 10.) Would you say the relationship between the bound and the number of training samples is the same in theory and in the experiment, i.e., do the plots have similar shapes? If not, conjecture why if you can.

d) It has been claimed that even when the theoretical generalization bounds are too loose to be directly useful (e.g., ε_{VC} or $\varepsilon_{eff} \geq 1$), the functional dependencies on the parameters still generally apply. To test this claim in an extreme case, set $N_{in} = 10$, and repeat (a) and (b) (except using $N_{in} = 10$ throughout). Then do this again for $N_{in} = 100$.

2. ***Dataset methodolgy and data-flow diagrams*** [created by Keith Jenkins]

Note: this is a theory and pencil-and-paper problem. No computer runs are necessary.

You are designing and testing a machine learning system, starting with the following steps:

(i) You do some initial tests and investigations with a pre-training set, and then discard the pre-training set. Let $\mathcal{D}$ be the remainder of the dataset.

(ii) You separate $\mathcal{D}$ into disjoint sets $\mathcal{D}'$, $\mathcal{D}_{\text{val}}$, and $\mathcal{D}_{\text{test}}$.

(a) Next, you want to do the following; the ordering and possible combining of these steps is up to you.

You want to try 3 different normalization techniques, and see which gives the best results. So you try standardization, min-max normalization ($\hat{x}_j = \frac{x_j - \min(x_j)}{\max(x_j) - \min(x_j)}$), and binarization ($\hat{x_j} = \text{sign}(x_j - \mu_j)$). For each, you train a linear Bayes classifier so that you can choose the normalization technique that gives the lowest error.

You also want to perform feature-space dimensionality adjustment – reducing the number of features *d* from the original feature set (D features) down to 1 feature, in increments of 1, to find an optimal value of *d*. The algorithm you use for feature reduction trains a 1-tree CART for each value of *d*. (It also chooses which *d* features are the best to use.)

There will be subsequent steps for trying and comparing different learning algorithms and hypothesis sets, but this part (a) focuses on the normalization and feature-space dimensionaltiy adjustment steps.

(i) Describe how you would use the datasets to perform the tasks outlined above. Be specific, so that it is clear which datasets are used for which tasks (and which part of which tasks), and in what order.

(ii) Draw a data-flow diagram (similar to diagrams on p. 15 and p. 19 of Lecture 10 notes) to illustrate how you would use the datasets to perform these tasks.

Comments: (1) There is no cross-validation in this part (a). However, you may further divide the given datasets into subsets if you deem it appropriate (if you do, be sure to define your new datasets clearly). (2) The scoring for this problem includes how good your data handling is.

(b) Separately from part (a), you are going to do model selection with training and cross validation.

(i) For the example of dataset usage for model selection in Lecture 10 (Example 1, page 11 and subsequent pages), draw a data-flow diagram (similar to diagrams on p. 15 and p. 19 of Lecture 10 notes), showing how you can perform model selection using *K*-fold cross validation. Learning will also be part of your data-flow diagram. Your diagram should show the dataset usage and dataflow for the combined procedure.

As part of the procedure shown in your diagram, obtain the validation-error estimate E_{CV} as the average over the *K* folds for each value of the parameter λ: $\mathrm{E_{CV}}(\mathrm{g_m}) = \frac{1}{\mathrm{K}}\sum_{\mathrm{k=1}}^{\mathrm{K}} \mathrm{E}_{\mathcal{D}_\mathrm{k}}\left(\mathrm{g_m^{(k)}}\right)$ in which *m* is the index of the λ parameter.

Also, after the model selection/cross validation procedure is finished, use the best model parameter λ_{m*}, re-train using $\mathcal{D}_{Tr\_final} = \mathcal{D}_{val} \cup \mathcal{D}'$, and test to get the error estimate $E_{\mathrm{test}}(g_m *)$. Include this in your diagram, or make a separate diagram showing it.

Tip: there may be more than one way to set up some of these procedures. Techniques that have the best dataset usage will get the most points.

(ii) Let the number of data points in one training dataset, and one validation set, in the kth fold of the cross-validation procedure, be $N_{Tr}^{(k)}$ and $N_{Val}^{(k)}$, respectively.

For the same procedure as (b)(i), give the generalization error bound ($\varepsilon_{\mathrm{VC}}$ or ε_M) for $E_{\mathcal{D}_k}\left(g_m^{(k)}\right)$. If you need other values, define and use a variable.

Also, give the generalization error bound on your final error $E_{\mathrm{test}}(g_m *)$.

3. ***Regression to estimate comment volume on Facebook.*** [Created by Zhiruo Zhou]

You are given a dataset adapted from the Facebook Comment Volume Dataset Data Set on UCI repository:
https://archive.ics.uci.edu/ml/datasets/Facebook+Comment+Volume+Dataset#

You may take a look at the website if you're interested in details about features and the background. For this problem, please directly use the npz file provided in the course material.
Three sets are included in the npz file: the training set, the validation set and the testing set, which contains 27435, 13514 and 10044 samples, respectively. The feature dimension of each sample is 53, and all features are numerical values. The target value to regress is a non-negative integer. In this problem, you don't need to round your float estimation to integers, and any evaluation such as MSE can directly be calculated using your float estimation and the ground truth.

You'll try the following regression models on the real-world data and conduct some analysis: linear regression, linear regression with l_1/l_2 regularization, CART, random forest and Adaboost.
For all model selection process, please use the training set to train models and use the validation set to select the final one. After selection, just test with the selected model and no need to train on (train+val). Don't standardize the dataset unless specified in questions.

Programming questions (for parts (b)-(f), see the "Hints on functions to use" box below):

(a) Come up with a baseline regressor which doesn't use much learning for later comparison. Explain your idea and why it would be a reasonable baseline. Implement your baseline regressor and report the testing MSE.
Note: your baseline should still use (or learn from) the training data but not much learning. This means that a baseline system that randomly generates an arbitrary number as output, without taking consideration of the dataset, would not qualify. The baseline cannot be one of the regression models mentioned in the problem description (such as linear regression), or a model of similar complexity.

(b) Try linear regression model with three regularization settings: no regularization, l_1 regularization and l_2 regularization. As for the regularization coefficient λ, search from $\log_2 \lambda = -10$ to $\log_2 \lambda = 10$ at step size of $\Delta \log_2 \lambda = 1$. Choose the parameter with the best val_MSE. Report the val_MSE and test_MSE of your best models.
Table you can use:

	Best λ	val_MSE	test_MSE
No regularization	-		
l_1 regularization			
l_2 regularization			

(c) Standardize all data in the following way: calculate the mean and std of all feature dimensions on the training set, and then subtract the mean and divide by the std to standardize (train, val, test). Repeat (b). Present the table and answer:
i. Do you observe any difference on the learned coefficients among three models? Explain.
ii. Compare test_MSE with those in (b). Which methods have obvious changes and which not? Explain.

Note: there might be some features which have the same value for all samples. Please handle their std values properly so there won't be divide-by-zero errors. Standardization only applies for this question, and don't do this in other questions.

(d) Try CART. Search from 1 to 10 on max_depth. As for other parameter setting, please use (criterion='mse', max_features=None, random_state=0) and default for others. Report the val_MSE and test_MSE of your best model.

(e) Try random forest. Set max_depth as the best one you find in (d). Search from 2 to 30 on n_estimators. Report the val_MSE and test_MSE of your best model.

(f) Try Adaboost. Set max_depth and n_estimators as your best values in previous questions. Search for a good value for learning_rate. Report the val_MSE and test_MSE of your best model.

(g) Do those regressors learn from the data compared with the baseline? Compare and comment on (explain) their performances relative to the baseline, and relative to each other.

Written questions (no computer simulations are necessary in (h), (i) below):

(h) Some of the features only have 0/1 values because they are converted from categorical features. Now for some of those categorical features, you modify the converted numerical values to 0/12345 (i.e., that attribute is either 0 or 12345) for all data including training, val and test and then do the regression problem again with the same random seed. Will your regressor give different estimations compared to your previous results? Give Yes/No answers for linear regression, CART, random forest and Adaboost, and explain why.

Note: Assume that the change of value doesn't affect any random process such as the feature sampling in random forest.

(i) What's the underlying assumption for the prior distribution of the model parameter in linear regression with l_1 regularization? Now given that the prior distribution is Exponential, derive the regularization term you should use. You may assume the components of $\underline{w}$ are independent for the purpose of the regularizer term.

Also answer: how could you ensure the w_j will all be positive in this problem?

Note: if a random variable x is Exponential(θ), then its probability density function is $p(x) = \frac{1}{\theta}\exp\left(-\frac{x}{\theta}\right), \theta > 0, x \geq 0$.

Hints on functions to use:

```
sklearn.linear_model. LinearRegression
sklearn.linear_model. Lasso
sklearn.linear_model. Ridge
sklearn.tree.DecisionTreeRegressor
sklearn.ensemble.RandomForestRegressor
sklearn.ensemble.AdaBoostRegressor
```