CS5785 Homework 1

The homework is generally split into programming exercises and written exercises.

This homework is due on **October 1, 2020 at 11:59 PM ET**. Upload your homework to Gradescope (Canvas->Gradescope). There are two assignments for this homework in Gradescope. Please note a complete submission should include:

1. A write-up as a single `.pdf` file. ⟶ Submit to "Homework 1- Write Up".
2. Source code for all of your experiments (AND figures) zipped into a single .zip file, in `.py` files if you use Python or `.ipynb` files if you use the IPython Notebook. If you use some other language, include all build scripts necessary to build and run your project along with instructions on how to compile and run your code. **If you use the IPython Notebook to create any graphs, please make sure you also include them.** ⟶ Submit to "Homework 1- Code".

The write-up should contain a general summary of what you did, how well your solution works, any insights you found, etc. On the cover page, include the class name, homework number, and team member names. You are responsible for submitting clear, organized answers to the questions. You could use online LaTeX templates from Overleaf, under "Homework Assignment" or "Project / Lab Report".

Please include all relevant information for a question, including text response, equations, figures, graphs, output, etc. If you include graphs, be sure to include the source code that generated them. Please pay attention to Canvas for relevant information regarding updates, tips, and policy changes. You are encouraged (but not required) to work in groups of 2.

If you need help

There are several strategies available to you.

- If you get stuck, we encourage you to post a question on the Discussions section of Canvas. That way, your solutions will be available to other students in the class.
- The professor and TAs offer office hours, which are a great way to get some one-on-one help.
- You are allowed to use well known libraries such as `scikit-learn`, `scikit-image`, `numpy`, `scipy`, etc. for this assignment. Any reference or copy of public code repositories should be properly cited in your submission (examples include Github, Wikipedia, Blogs).

PROGRAMMING EXERCISES

1. Digit Recognizer

   (a) Join the Digit Recognizer competition on Kaggle. Download the training and test data. The competition page describes how these files are formatted.

   (b) Write a function to display an MNIST digit. Display one of each digit.

   (c) Examine the prior probability of the classes in the training data. Is it uniform across the digits? Display a normalized histogram of digit counts. Is it even?

   (d) Pick one example of each digit from your training data. Then, for each sample digit, compute and show the best match (nearest neighbor) between your chosen sample and the rest of the training data. Use L_2 distance between the two images' pixel values as the metric. This probably won't be perfect, so add an asterisk next to the erroneous examples (if any).

   (e) Consider the case of binary comparison between the digits 0 and 1. Ignoring all the other digits, compute the pairwise distances for all genuine matches and all impostor matches, again using the L_2 norm. Plot histograms of the genuine and impostor distances on the same set of axes.

   (f) Generate an ROC curve from the above sets of distances. What is the equal error rate? What is the error rate of a classifier that simply guesses randomly?

   (g) Implement a K-NN classifier. (You cannot use external libraries for this question; it should be your own implementation.)

   (h) Take 15% of your dataset and make it your holdout set. Train the classifier on the remaining data, using different values of K and use the holdout set to determine the best value of K. Print a table showing the values of K you tried as well as their training and holdout accuracy. Report the best value of K that you found.

   (i) Generate a confusion matrix (of size 10×10) from your results. Which digits are particularly tricky to classify?

   (j) Train your classifier with all of the training data, and test your classifier with the test data. Submit your results to Kaggle.

2. Predicting House Prices

   (a) Join the House Prices competition on Kaggle. Download the training and test data.

   (b) Using least squares, try to predict house prices on this dataset. You can choose the features (or combinations of features) you would like to use or ignore, provided you justify your reasoning.

   (c) Now try predicting house prices using regularized least squares. Try both L1 and L2 regularization and compare them. Briefly describe your findings from training regularized and unregularized models.

   (d) Train your model using all of the training data, and test it using the testing data. Submit your results to Kaggle.

WRITTEN EXERCISES

1. Maximum Likelihood and KL Divergence. Let $\hat{p}(x, y)$ denote the empirical data distribution over a space of inputs $x \in \mathcal{X}$ and outputs $y \in \mathcal{Y}$. For example, in an image recognition task, x can be an image and y can be whether the image contains a cat or not. Let $p_\theta(y|x)$ be a probabilistic classifier parameterized by θ, e.g., a logistic regression classifier with coefficients θ. Show that the following equivalence holds:

$$\arg\max_{\theta\in\Theta} \mathbb{E}_{\hat{p}(x,y)}\left[\log p_\theta(y|x)\right] = \arg\min_{\theta\in\Theta} \mathbb{E}_{\hat{p}(x)}\left[\mathrm{KL}(\hat{p}(y|x)\,\|\,p_\theta(y|x))\right].$$

where KL denotes the KL-divergence:

$$\mathrm{KL}(p(x)\,\|\,q(x)) = \mathbb{E}_{x\sim p(x)}[\log p(x) - \log q(x)].$$

2. Bayes rule for quality control. You're the foreman at a factory making ten million widgets per year. As a quality control step before shipment, you create a detector that tests for defective widgets before sending them to customers. The test is uniformly 95% accurate, meaning that the probability of testing positive given that the widget is defective is 0.95, as is the probability of testing negative given that the widget is not defective. Further, only one in 100,000 widgets is actually defective.

   (a) Suppose the test shows that a widget is defective. What are the chances that it's actually defective given the test result?

   (b) If we throw out all widgets that are test defective, how many good widgets are thrown away per year? How many bad widgets are still shipped to customers each year?

3. In k-nearest neighbors, the classification is achieved by majority vote in the vicinity of data. Suppose our training data comprises n data points with two classes, each comprising exactly half of the training data, with some overlap between the two classes.

   (a) Describe what happens to the average 0-1 prediction error on the training data when the neighbor count k varies from n to 1. (In this case, the prediction for training data point x_i includes (x_i, y_i) as part of the example training data used by kNN.)

   (b) We randomly choose half of the data to be removed from the training data, train on the remaining half, and test on the held-out half. Predict and explain with a sketch how the average 0-1 prediction error on the held-out validation set might change when k varies? Explain your reasoning.

   (c) In kNN, once k is determined, all of the k-nearest neighbors are weighted equally in deciding the class label. This may be inappropriate when k is large. Suggest a modification to the algorithm that avoids this caveat.