Machine Learning	Homework 1
CSCI 5622 Fall 2017	Due Time Sep 15, 2017

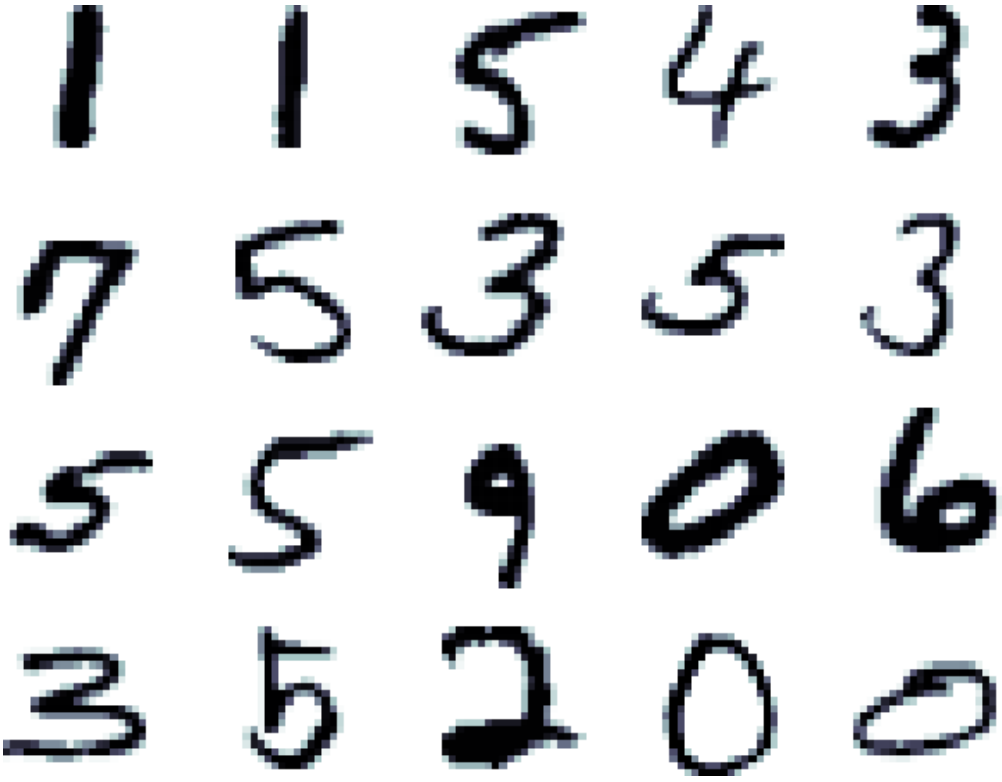
Submission Guidelines

- In order to download files required for the homework, clone `https://github.com/BoulderDS/csci_5622_hws`.
- For programming questions, submit python source files in a zip file.
- For other questions, submit a PDF file of no more than 4 pages.

All homework submissions are done through Moodle.

1 K-nearest Neighbor (40pts)

In this homework you'll implement a k-Nearest Neighbor classification framework to take a drawing of a number and output what number it corresponds to.



This question is designed to be an easy homework to get you familiar with the basic workflow in machine learning. If you're spending a lot of time on this question, you are either:

- not prepared to take the course (i.e., if you're struggling with Python)
- seriously over-thinking the assignment

- trying to reimplement too much of the assignment

Most of this assignment will be done by calling libraries that have already been implemented for you. If you are implementing n-dimensional search or a median algorithm, you are generating extra work for yourself and making yourself vulnerable to errors.

1.1 Programming questions (30 pts)

Finish `knn.py` to achieve the following goals. You can use `tests.py` to test your code.

1. Store necessary data in the constructor so you can do classification later
2. Modify the *majority* function so that it returns the **value** associated with the most **indices**.
3. Modify the *classify* function so that it finds the closest indicies to the query point.
4. Modify the *confusion matrix* function to classify examples and record which number it got right.

1.2 Analysis (5 points)

1. What is the role of the number of training instances to accuracy (hint: try different "`--limit`" and plot accuracy vs. number of training instances)?
2. What numbers get confused with each other most easily?
3. What is the role of k to training accuracy?
4. In general, does a small value for k cause "overfitting" or "underfitting"?

2 Cross Validation (30pts)

In practice, we usually use cross validation to find the best hyperparameter (k in $k-$nearest neighbor). In this part of homework, we will implement cross validation to evaluate the best parameter on the training data. Specifically, we split the training randomly into K folds. For a parameter k, for each $i = 1, \ldots, K$, we use the i-th fold to evaluate the performance of the classifier from the other $K - 1$ folds. We then average the performance across all K folds.

2.1 Programming questions (20pts)

Finish `cross_validation.py` to conduct K-fold cross-validation on the MNIST data and find the best $k \in \{1, 3, 5, 7, 9\}$. Since the cross validation can get slow, run it with "`--limit 500`" and "`--limit 5000`".

2.2 Analysis (10pts)

1. What is the best k chosen from 5-fold cross validation with "`--limit 500`"?
2. What is the best k chosen from 5-fold cross validation "`--limit 5000`"?
3. Is the best k consistent with the best performance k in problem 1?

3 Bias-variance tradeoff (20pts)

Derive the bias-variance decomposition for k-NN regression in class. Specifically, assuming the training set is fixed $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where the data are generated from the following process $y = f(x) + \epsilon, \mathrm{E}(\epsilon) = 0, \mathrm{Var}(\epsilon) = \sigma_\epsilon^2$. k-NN regression algorithm predict the value for x_0 as $h_S(x_0) = \frac{1}{k}\sum_{l=1}^{k} y_{(l)}$, where $x_{(l)}$ is the $l-$th nearest neighbor to x_0. $\mathrm{Err}(x_0)$ is defined as $\mathrm{E}((y_0 - h_S(x_0))^2)$.

Prove that

$$\mathrm{Err}(x_0) \quad = \quad \sigma_\epsilon^2 + \big[f(x_0) - \frac{1}{k}\sum_{l=1}^{k} f(x_{(l)})\big]^2 + \frac{\sigma_\epsilon^2}{k}.$$

4 Syllabus quiz (5pts)

Read the syllabus (`https://chenhaot.com/courses/csci5622/2017fa/syllabus.html`) and the final project assignment (`https://chenhaot.com/courses/csci5622/2017fa/final-project.html`). Finish this online quiz: `https://goo.gl/forms/Cpn8oIB5GMBBAFmc2`