Machine Learning
CSCI 5622 Fall 2017
Name: Chi Chen

Homework 1
Due Time Sep 15, 2017
CU identity key: chch4713

# 1 K-nearest Neighbor (40pts)

I. What is the role of the number of training instances to accuracy (hint: try different "--limit" and plot accuracy vs. number of training instances)?

The larger the number of training instances are, the higher accuracy can be . When the number of training instances are 500, the accuracy is 83.11%. And when the training instances up to 5000, the accuracy increases to 94.01%. Furthermore, if you use all training set to train the model, the accuracy will be 97.27%. (All data based on k = 3)

II. What numbers get confused with each other most easily?

```
chenqis-MacBook-Pro:hw1 chenchi$ python knn.py
<__main__.Numbers object at 0x110734f60>
Done loading data
        0       1       2       3       4       5       6       7       8       9
------------------------------------------------------------------------------------------
0:      984     0       2       0       0       0       2       0       1       2
1:      0       1060    1       0       1       0       1       1       0       0
2:      4       8       953     2       1       1       1       19      1       0
3:      0       0       7       1002    0       8       1       3       6       3
4:      0       11      0       0       951     0       0       2       0       19
5:      2       0       2       20      1       869     15      2       1       3
6:      1       1       0       0       0       1       964     0       0       0
7:      0       10      0       0       3       0       0       1071    0       6
8:      6       8       3       10      4       16      4       4       947     7
9:      3       3       1       10      14      3       0       6       2       919
Accuracy: 0.972000
```

Based on the result above, the most easily numbers to get confused are set (2,7) and set (4,9) .

III. What is the role of k to training accuracy?

| | Limit = 500 | Limit = 5000 |
|---|---|---|
| k = 1 | 84.58% | 93.88% |
| k = 3 | 83.11% | 94.01% |
| k = 5 | 79.95% | 93.29% |
| k = 7 | 79.63% | 93.03% |

Since the difference of size of data set, the best k will be different too. It's hard to say the large k will be better or the smaller k will be better in this set. But when the data set is getting bigger, the best k will be a little bit larger too. The small k will cause the overfitting.

IV. In general, does a small value for k cause “overfitting” or “underfitting”?

In general, if the data set is large enough, a small k will cause “overfitting”, because it took only one nearest data as itself’s label. The bias will be low but the variance will be high.

## 2 Cross Validation (30pts)

I. What is the best k chosen from 5-fold cross validation with “--limit 500”?

The best k chosen from 5-fold cross validation with “—limit 500” is 3.

II. What is the best k chosen from 5-fold cross validation “--limit 5000”?

The best k chosen from 5-fold cross validation with “—limit 5000” is 1.

III. Is the best k consistent with the best performance k in problem 1?

No, the best k didn’t consistent with the best performance k with only KNN. The best k for “—limit 500” with only KNN is 1 but the best k with KNN and CV is 3. And The best k for “—limit 5000” with only KNN is 3 but the best k with KNN and CV is 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

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

$$
\begin{aligned}
Err(x_0) &= E((y_0 - h_s(x_0))^2) \\
&= \sigma_\varepsilon{}^2 + [Eh(x_0) - f(x_0)]^2 + E[h(x_0) - Eh(x_0)]^2 \\
&= \sigma_\varepsilon{}^2 + [f(x_0) - Eh(x_0)]^2 + Var(h(x_0)) \\
&= \sigma_\varepsilon{}^2 + [f(x_0) - E\frac{1}{k}\sum_l^k y(l)]^2 + Var(\frac{1}{k}\sum_l^k y(l)) \\
&= \sigma_\varepsilon{}^2 + [f(x_0) - E\frac{1}{k}\sum_l^k (f(x_{(l)}) + \varepsilon)]^2 + Var(\frac{1}{k}\sum_l^k (f(x_{(l)}) + \varepsilon)) \\
&= \sigma_\varepsilon{}^2 + [f(x_0) - E\frac{1}{k}\sum_l^k (f(x_{(l)})) + E\frac{1}{k}\sum_l^k (\varepsilon)]^2 + Var(\frac{1}{k}\sum_l^k (f(x_{(l)})) + \frac{1}{k}\sum_l^k (\varepsilon)) \\
&= \sigma_\varepsilon{}^2 + [f(x_0) - E\frac{1}{k}\sum_l^k (f(x_{(l)})) + 0]^2 + Var(0 + \frac{1}{k}\sum_l^k (\varepsilon)) \\
&= \sigma_\varepsilon{}^2 + [f(x_0) - \frac{1}{k}\sum_l^k E(f(x_{(l)}))]^2 + \frac{1}{k^2}\sum_l^k Var(\varepsilon) \\
&= \sigma_\varepsilon{}^2 + [f(x_0) - \frac{1}{k}\sum_l^k (f(x_{(l)}))]^2 + \frac{1}{k}\sigma_\varepsilon{}^2
\end{aligned}
$$