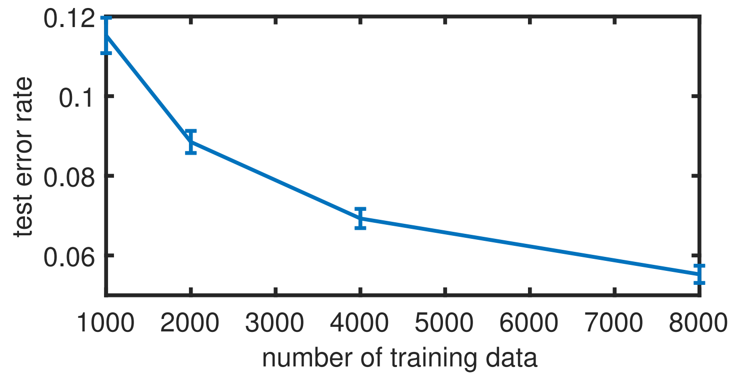
COMS 4771 Fall 2016 Homework 1

Solutions by Daniel Hsu

Problem 1

Solution. Learning curve plot (error bars extend one standard deviation below the mean to one standard deviation above the mean).



```
function preds = nn(X,Y,test)
[~,I] = max(bsxfun(@minus,2*test*X',dot(X,X,2)'),[],2);
preds = Y(I);
```

```
import numpy as np
def nn(X,Y,test):
  return Y[np.argmin(np.sum(np.square(X),axis=1)-2*test.dot(X.T),axis=1)]
```

□

Problem 2

Solution. There are many ways to do prototype selection; the solution here is just one possible heuristic. It is by no means the best method or even necessarily based on formal principles. Possibly its only virtue is that it is relatively easy to describe and implement.

1. Partition $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ into training and validation sets (with uniformly random $(80\%, 20\%)$ split). For each class $k \in \{1, 2, \dots, K\}$, compute the nearest neighbor among the training set examples with label k for each point in the validation set with label k, and select the m/K points that are chosen as the nearest neighbor the most often.

2. Below, we use the notation $\mathrm{NN}(\boldsymbol{q}, S)$ to denote the nearest neighbor of $\boldsymbol{q}$ among points in S.

```
Input: labeled examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, positive integer $m$.
1. Partition examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ uniformly at random into training set
   $S$ (with $|S| = \lfloor 0.8n \rfloor$) and validation set $T$.
2. Initially $P = \emptyset$.
3. For each class $k \in \{1, 2, \dots, K\}$:
   3.1. Let $S_k := \{\boldsymbol{x} : (\boldsymbol{x}, k) \in S\}$ and $T_k := \{\boldsymbol{q} : (\boldsymbol{q}, k) \in T\}$, the points in
        $S$ and $T$, respectively, that have label $k$.
   3.2. Let $N_k := \{\mathrm{NN}(\boldsymbol{q}, S_k) : \boldsymbol{q} \in T_k\}$, the mapping of all $\boldsymbol{q} \in T_k$ to
        $\mathrm{NN}(\boldsymbol{q}, S_k)$.
   3.3. Select $\lfloor m/K \rfloor$ points $\boldsymbol{x} \in S_k$ that appear most often in $N_k$; add
        such $(\boldsymbol{x}, k)$ to $P$.
4. Return $P$.
```

3. The table below shows the average (and standard deviation) of the test error rates across 10 repeated trials.

$m = 1000$	$m = 2000$	$m = 4000$	$m = 8000$
0.0934 (0.0020)	0.0730 (0.0023)	0.0593 (0.0010)	0.0480 (0.0014)

This method seems to give a small improvement over random selection of prototypes.

□

Problem 3

Solution.

(a) Let X and Y be independent and uniformly distributed $\mathcal{C}$-valued random variables. Then

$$\mathbb{P}(X = Y) = \sum_{c \in \mathcal{C}} \mathbb{P}(X = c, Y = c) = \sum_{c \in \mathcal{C}} \mathbb{P}(X = c) \cdot \mathbb{P}(Y = c) = \sum_{c \in \mathcal{C}} \left(\frac{n_c}{100} \right)^2 ,$$

where the first equality follows from the law of total probability, and the second equality follows from the independence of X and Y. (This is called the *collision probability*.) Therefore, the probability that the two picks balls have different colors is $1 - \mathbb{P}(X = Y) = 1 - \sum_{c \in \mathcal{C}} (n_c/100)^2$.

(b) We recognize the probability in (a) as the Gini index for a probability distribution over $\mathcal{C}$ with probabilities $(n_c/100 : c \in \mathcal{C})$. We know from lecture that this is maximized when the distribution is uniform. (This can also be easily proved using the Cauchy-Schwarz inequality.) So we should paint the balls so that there are 20 balls of each color.

□