Homework 1

Hao Xu
hx2208

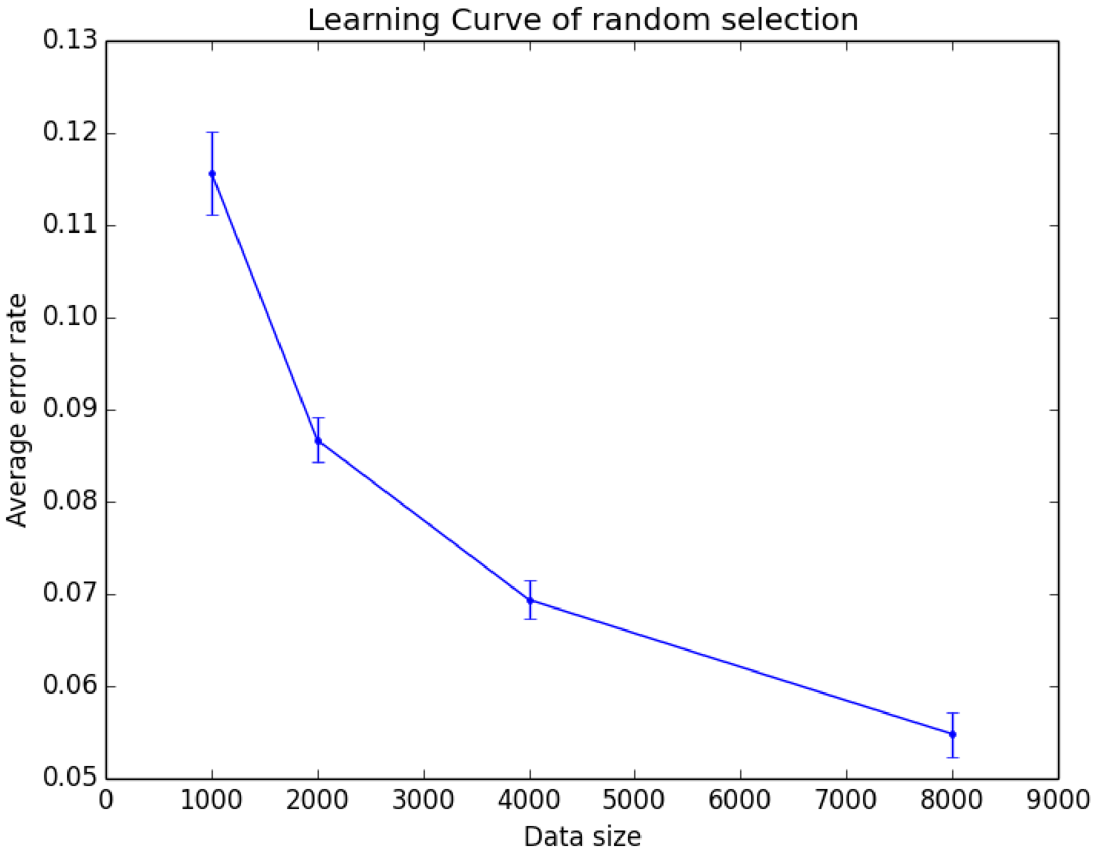
Problem 1.

Answer:

Running result for 10 trails:

Trials	1000	2000	4000	8000
1	0.1112	0.0867	0.0711	0.0546
2	0.1183	0.086	0.0715	0.0538
3	0.1174	0.0882	0.0701	0.0592
4	0.1097	0.0882	0.0682	0.0546
5	0.1125	0.083	0.0678	0.0509
6	0.1231	0.0842	0.0655	0.0537
7	0.1126	0.0853	0.0673	0.0565
8	0.1216	0.0917	0.0696	0.0571
9	0.1151	0.0865	0.0719	0.0519
10	0.1152	0.0871	0.0707	0.0556
average	0.11567	0.08669	0.06937	0.05479
stdev	0.00442	0.00241	0.00209	0.00245

The learning curve plot is:



Problem 2.

Answer:

1. Description of prototype selection:
My prototype data consists of two parts. Firstly, $\frac{1}{2}$ of the prototype data is chosen directly from the training data. Each label has equal number of times to be chosen, which are $\frac{1}{20}$ m. This set is going to be used as the initial seed set.
Then divide the whole training data set into 10 parts. For each part of the training, use the seed set to make prediction on the training subset. Calculate the rows that are not correctly predicted. Randomly select $\frac{1}{20}$ m rows and insert them into seed set. (The reason to choose $\frac{1}{2}$ m is we have divided the whole train into 10 parts and each parts choose $\frac{1}{20}$ m which gives the rest $\frac{1}{2}$ m)
2. Pseudocode of the algorithm:

I used a function I wrote in problem 1 to do classification in problem 2. So in pseudocode I would assume this function is written and I could call it directly.

Function classifier(train, label, test), which returns a vector of prediction labels.
Function prototypeSelector(train, label, m):

```
$PrototypeSet \leftarrow \emptyset$
$PrototypeLabelSet \leftarrow \emptyset$
$labelcounters[10] \leftarrow [0, ..., 0]$
for $i = 0$ to $len(train)$ do
    if $labelcounters[label[i]] < \frac{1}{20}m$ then
        $PrototypeSet \leftarrow Prototypes \cup train[i]$
        $PrototypeLabelSet \leftarrow Prototypes \cup label[i]$
        $labelcounters[label[i]] \leftarrow labelcounters[label[i]] + 1$
    end if
end for
$TrainCuts \leftarrow [0, ..., 0]$    ▷ To divide train set into 10 parts
for $i = 0$ to $11$ do
    $TrainCuts[i] \leftarrow \frac{i}{10} len(train)$
end for
for $i = 0$ to $10$ do
    $pred \leftarrow classifier(PrototypeSet, PrototypeLabelSet, train[TrainCuts[i] : TrainCuts[i+1]])$    ▷ Use prototypeset to make prediction on each sub training set
    $diff \leftarrow pred - label[TrainCuts[i] : TrainCuts[i+1]]$    ▷ Compare labels
    $indeces \leftarrow random(diff) + offsets$    ▷ randomly select indeces from the labels that are different. Ignore the specific way to do in pesudocode, offsets is the starting cut for each 1/10 training set
    $PrototypeSet \leftarrow PrototypeSet \cup train[indeces]$
    $PrototypeLabelSet \leftarrow PrototypeLabelSet \cup label[indeces]$
end for
return $PrototypeSet, PrototypeLabelSet$
```

3. Table of results:

Trials	1000	2000	4000	8000
1	0.104	0.0844	0.0692	0.0564
2	0.1071	0.0876	0.0709	0.0572
3	0.1083	0.0851	0.0676	0.0564
4	0.1081	0.0854	0.0697	0.0574
5	0.1138	0.0836	0.0672	0.0579
6	0.1114	0.0833	0.0668	0.0571
7	0.1047	0.0852	0.069	0.0563
8	0.1131	0.0822	0.0662	0.0567
9	0.1108	0.0821	0.0685	0.0571
10	0.1091	0.0831	0.0666	0.0564
average	0.10904	0.0842	0.06817	0.05689
stdev	0.003287	0.001688	0.001533	0.0005343

Problem 3.

Answer:

(a) The probability of getting one same color is:

$(\frac{n_c}{100})^2$, for each $c \in \{\text{red}, \text{orange}, \text{yellow}, \text{green}, \text{blue}\}$

Thus, the probability of getting a same color is $\Sigma(\frac{n_c}{100})^2$, for all $c \in \{\text{red}, \text{orange}, \text{yellow}, \text{green}, \text{blue}\}$

Thus, the probability of getting different color is:

$1 - \Sigma(\frac{n_c}{100})^2$, for all $c \in \{\text{red}, \text{orange}, \text{yellow}, \text{green}, \text{blue}\}$

(b) Expand the formula in (a), get

$P = 1 - ((\frac{n_{red}}{100})^2 + (\frac{n_{orange}}{100})^2 + (\frac{n_{yellow}}{100})^2 + (\frac{n_{green}}{100})^2 + (\frac{n_{blue}}{100})^2)$

And, $n_{red} + n_{orange} + n_{yellow} + n_{green} + n_{blue} = 100$

In order to maximize P, need to minimize $\Sigma(\frac{n_c}{100})^2$

Let $\Phi = (\frac{n_{red}}{100})^2 + (\frac{n_{orange}}{100})^2 + (\frac{n_{yellow}}{100})^2 + (\frac{n_{green}}{100})^2 + (\frac{n_{blue}}{100})^2 + \lambda(n_{red} + n_{orange} + n_{yellow} + n_{green} + n_{blue} - 100)$

According to method of Lagrange multipliers,

get equations:

$$\begin{cases} 2n_{red} + \lambda = 0 \\ 2n_{orange} + \lambda = 0 \\ 2n_{yellow} + \lambda = 0 \\ 2n_{green} + \lambda = 0 \\ 2n_{blue} + \lambda = 0 \\ n_{red} + n_{orange} + n_{yellow} + n_{green} + n_{blue} = 100 \end{cases} \tag{1}$$

According to the equations, get $n_{red} = n_{orange} = n_{yellow} = n_{green} = n_{blue}$

Thus, $n_{red} = n_{orange} = n_{yellow} = n_{green} = n_{blue} = 20$.

The probability is: $1 - 5 * (\frac{1}{5} * \frac{1}{5}) = \frac{4}{5}$