Machine Learning - Homework 1

Parita Pooj (psp2133)

September 21, 2016

Problem 1

Code provided in the zip folder with the plot for Learning Curve.

Problem 2

(a) **Description:**
For prototype selection, it would beneficial to combine similar datapoints and have a representative of each of these combination in the smaller training data. To do this, we can use techniques like K-means or meanshift to find the representative points. While meanshift is more reliable, it is more complex and can select arbitrary number of data points. For this problem, I have used K-Means to find 1000 clusters from the set of 60,000 data points where the centroid becomes the representative data points of each cluster. The label for this centroid is assigned as the label which gets majority votes in the cluster. The K-means function for MATLAB is included in the zip, and it is the implementation by Tim Benham for Fast K-means algorithm.

(b) **Pseudocode:**

---

1. Load from ocr matrix
2. Divide the data points into 10 sets $D_l \in D$ based on their labels where each label, $l \in 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$
3. Run K-means[1] clustering algorithm on each set,
cluster labels, cluster centers $\leftarrow$ kmeans(data, $m/10$) to form $m/10$ clusters
4. For each cluster center, find the majority label of the cluster of data points.
5. New trainind data and labels is combined from cluster centers and majority labels from each D_l
$ndata, nlabels \leftarrow \cup_{D_l \in D}$ cluster centers, majority label

---

(c) **Test Error Rates:**

m	1000	2000	4000	8000
Error Rates in %	4.08	3.95	3.29	2.89

Problem 3

(a) Let the Probability of picking two balls of different color from the urn with replacement be P.
Let P_c = P(ball 2 is not of color c $\mid$ ball 1 is of color c)
$P = \sum_{c \in C} P_c$
Since, $P_c = \frac{(n_c)(100-n_c)}{n}$
$P = \sum_{c \in C} \frac{(n_c)(100-n_c)}{n}$

(b) We want to maximize P given that $\sum_{c \in C} n_c = 100$
For each n_c, $\frac{\partial(P+\lambda(100-n_c))}{\partial n_c} = 0$
We get the similar equations for all n_c, which gives us:
$n_r = n_o = n_y = n_g = n_b$
Thus, probability will be maximum when we have equal number of balls for every color.
For $n = 100$, $n_c = 20$ for all $c \in C$

References

[1] fkmeans MATLAB function, https://www.mathworks.com/matlabcentral/fileexchange/31274-fast-k-means