E0 270 Machine Learning *Due:* Jan 24, 2012

Assignment 1 - Report

Patel Kevin Niranjanbhai *04-04-00-10-41-11-1-08262*

1 Solution: Problem 1

Let $h_\theta(x)$ be the hypothesis learnt by the classifier. Then let $err(x)$ be defined as follows:

$$err(x) = \begin{cases} m & \text{if } h_\theta(x) = 1 \text{ and } y = -1 \\ n & \text{if } h_\theta(x) = -1 \text{ and } y = 1 \\ 0 & \text{otherwise} \end{cases}$$

Then
E[$err(x)$] $= m.P(\ h_\theta(x) = 1 \text{ and } y = -1) + n.P(\ h_\theta(x) = -1 \text{ and } y = 1)$

2 Solution: Problem 2

a) Refer to attached code.
b) Learning Curves for k = 5

../../../../../../home/cse11/kevin.patel/HW1/Problem_2/p2b.jpg

c) Learning Curves for k = 1

../../../../../../home/cse11/kevin.patel/HW1/Problem_2/p2c.jpg

As is clear from the graph, k = 1 is not a good parameter. Intuitively, this is true, since labelling a point based on just one neighbour is not fair enough. However the train error is 0, but its just because every point is nearest to itself.

d) Errors as function of k

../../../../../../home/cse11/kevin.patel/HW1/Problem_2/p2d.jpg

This plot shows that a large value of k is not going to help that much either. Intuitively, this also holds true. For instance, if we take k large enough, we are "neighbours" of every citizen of India, but that information is not useful to label us in anyway.

e)

Decision Boundary for k = 1

../../../../../../home/cse11/kevin.patel/HW1/Problem_2/p2e_k1.jpg

Decision Boundary for k = 49

../../../../../../home/cse11/kevin.patel/HW1/Problem_2/p2e_k49.jpg

Decision Boundary for k = 499

../../../../../../home/cse11/kevin.patel/HW1/Problem_2/p2e_k499.jpg

So, as is observed, the boundary was most complex in case of k = 1, and most smooth in case of k=499. However, this smoothness shows a strong bias, which may result in misclassification of new datapoints. Thus ideally, the value of k should be chosen somewhere midway.

3 Solution: Problem 3

Learning Curves for Handwritten Digit Recognition using k Nearest Neighbours (with k = 5)

../../../../../home/cse11/kevin.patel/HW1/Problem_3/p3_le

The following test images were misclassified by the model trained on full training data.

../../../../../home/[illegible]

4 Solution: Problem 4

a)

Decision Tree 1

../../../../../home/cse11/kevin.patel/HW1/Problem_4/tree1.png

Decision Tree 2

../../../../../home/cse11/kevin.patel/HW1/Problem_4/tree2.png b)

Initial entropy of the system = $-p_L^+ log(p_L^+) - (1-p_L^+)log(1-p_L^+) = 1$.

Lets consider the various partitions that can be created at the first node.

Consider the partition made by checking $x_1 <= \frac{1}{4}$.

This leads to 6 datapoints on the left side and 18 points on the right side.

So
$m_L = 6$
$m_R = 18$.
$m_L^+ = 3$
$m_R^+ = 9$.

Now $p_L^+ = \frac{m_L^+}{m_L} = \frac{3}{6} = \frac{1}{2}$

So entropy(left_side) $N_L = -p_L^+ log(p_L^+) - (1-p_L^+)log(1-p_L^+) = 1$

Similarly $p_R^+ = \frac{m_R^+}{m_R} = \frac{9}{18} = \frac{1}{2}$

So entropy(right_side) $N_R = -p_R^+ log(p_R^+) - (1-p_R^+)log(1-p_R^+) = 1$

Now, total entropy $N = \frac{m_L}{m}N_L + \frac{m_R}{m}N_R = 1 * \frac{6}{24} + 1 * \frac{18}{24} = 1$

Therefore, change in entropy = 0

Now consider the partition made by checking $x_1 <= \frac{1}{2}$.

This leads to 13 datapoints on the left side and 11 points on the right side.

So
$m_L = 13$
$m_R = 11$.
$m_L^+ = 6$
$m_R^+ = 6$.

Now $p_L^+ = \frac{m_L^+}{m_L} = \frac{6}{13}$

So entropy(left_side) $N_L = -p_L^+ log(p_L^+) - (1-p_L^+)log(1-p_L^+) = 0.9957$

Similarly $p_R^+ = \frac{m_R^+}{m_R} = \frac{6}{11}$

So entropy(right_side) $N_R = -p_R^+ log(p_R^+) - (1-p_R^+)log(1-p_R^+) = 0.9940$

Now, total entropy N = $\frac{m_L}{m}N_L + \frac{m_R}{m}N_R = 0.9957 * \frac{13}{24} + 0.9940 * \frac{11}{24} = 0.9949$

Therefore, change in entropy = 0.0050

Now consider the partition made by checking $x_1 <= \frac{3}{4}$.

This leads to 20 datapoints on the left side and 4 points on the right side.

So
$m_L = 20$
$m_R = 4$.
$m_L^+ = 8$
$m_R^+ = 4$.

Now $p_L^+ = \frac{m_L^+}{m_L} = \frac{8}{20}$

So entropy(left_side) $N_L = -p_L^+ log(p_L^+) - (1 - p_L^+) log(1 - p_L^+) = 0.9709$

Similarly $p_R^+ = \frac{m_R^+}{m_R} = \frac{4}{4}$

So entropy(right_side) $N_R = -p_R^+ log(p_R^+) - (1 - p_R^+) log(1 - p_R^+) = 0$

Now, total entropy N = $\frac{m_L}{m}N_L + \frac{m_R}{m}N_R = 0.9709 * \frac{20}{24} + 0 * \frac{4}{24} = 0.8091$

Therefore, change in entropy = 0.1908

Now calculating the entropies for split points along x_2.

Consider the partition made by checking $x_2 <= \frac{1}{4}$.

So
$m_L = 8$
$m_R = 16$.
$m_L^+ = 3$
$m_R^+ = 9$.

Now $p_L^+ = \frac{m_L^+}{m_L} = \frac{3}{8}$

So entropy(left_side) $N_L = -p_L^+ log(p_L^+) - (1 - p_L^+) log(1 - p_L^+) = 0.9544$

Similarly $p_R^+ = \frac{m_R^+}{m_R} = \frac{9}{16}$

So entropy(right_side) $N_R = -p_R^+ log(p_R^+) - (1 - p_R^+) log(1 - p_R^+) = 0.9886$

Now, total entropy N = $\frac{m_L}{m} N_L + \frac{m_R}{m} N_R = 0.9544 * \frac{8}{24} + 0.9886 * \frac{16}{24} = 0.9773$

Therefore, change in entropy = 0.0227

Now consider the partition made by checking $x_2 <= \frac{1}{2}$.

So
$m_L = 12$
$m_R = 12.$
$m_L^+ = 4$
$m_R^+ = 8.$

Now $p_L^+ = \frac{m_L^+}{m_L} = \frac{4}{12}$

So entropy(left_side) $N_L = -p_L^+ log(p_L^+) - (1 - p_L^+) log(1 - p_L^+) = 0.9183$

Similarly $p_R^+ = \frac{m_R^+}{m_R} = \frac{8}{12}$

So entropy(right_side) $N_R = -p_R^+ log(p_R^+) - (1 - p_R^+) log(1 - p_R^+) = 0.9183$

Now, total entropy N = $\frac{m_L}{m} N_L + \frac{m_R}{m} N_R = 0.9183 * \frac{12}{24} + 0.9183 * \frac{12}{24} = 0.9183$

Therefore, change in entropy = 0.0817

Finally considering the partition made by checking $x_2 <= \frac{3}{4}$.

So
$m_L = 8$
$m_R = 16.$
$m_L^+ = 5$
$m_R^+ = 7.$

Now $p_L^+ = \frac{m_L^+}{m_L} = \frac{5}{8}$

So entropy(left_side) $N_L = -p_L^+ log(p_L^+) - (1 - p_L^+) log(1 - p_L^+) = 0.9544$

Similarly $p_R^+ = \frac{m_R^+}{m_R} = \frac{7}{16}$

So entropy(right_side) $N_R = -p_R^+ log(p_R^+) - (1 - p_R^+) log(1 - p_R^+) = 0.9887$

Now, total entropy N = $\frac{m_L}{m} N_L + \frac{m_R}{m} N_R = 0.9544 * \frac{8}{24} + 0.9887 * \frac{16}{24} = 0.9773$

Therefore, change in entropy = 0.0227

Since, change in entropy is maximum at the split point $\frac{1}{2}$ along x_1,
The root node should be $x_1 <= \frac{1}{2}$

c) Now lets rework the previous problem, but using Gini Index instead of Entropy as a measure of impurity.

Initially the gini_index of system = $p^+(1-p^+) = \frac{1}{2}.\frac{1}{2} = \frac{1}{4}$.

Lets consider the various partitions that can be created at the first node.

Consider the partition made by checking $x_1 <= \frac{1}{4}$.

This leads to 6 datapoints on the left side and 18 points on the right side.

So
$m_L = 6$
$m_R = 18$.
$m_L^+ = 3$
$m_R^+ = 9$.

Now $p_L^+ = \frac{m_L^+}{m_L} = \frac{3}{6} = \frac{1}{2}$

So gini_index(left_side) $N_L = p_L^+(1-p_L^+) = 0.25$

Similarly $p_R^+ = \frac{m_R^+}{m_R} = \frac{9}{18} = \frac{1}{2}$

So gini_index(right_side) $N_R = p_R^+(1-p_R^+) = 0.25$

Now, total gini_index N = $\frac{m_L}{m}N_L + \frac{m_R}{m}N_R = 0.25 * \frac{6}{24} + 0.25 * \frac{18}{24} = 0.25$

Therefore change in gini_index = 0.

Now consider the partition made by checking $x_1 <= \frac{1}{2}$.

This leads to 13 datapoints on the left side and 11 points on the right side.

So
$m_L = 13$
$m_R = 11$.
$m_L^+ = 6$
$m_R^+ = 6$.

Now $p_L^+ = \frac{m_L^+}{m_L} = \frac{6}{13}$

So gini_index(left_side) $N_L = p_L^+(1-p_L^+) = 0.2485$

Similarly $p_R^+ = \frac{m_R^+}{m_R} = \frac{6}{11}$

So gini_index(right_side) $N_R = p_R^+(1 - p_R^+) = 0.2479$

Now, total gini_index $N = \frac{m_L}{m} N_L + \frac{m_R}{m} N_R = 0.25 * \frac{13}{24} + 0.2485 * \frac{11}{24} = 0.2482$

Therefore change in gini_index = 0.0018.

Now consider the partition made by checking $x_1 <= \frac{3}{4}$.

This leads to 20 datapoints on the left side and 4 points on the right side.

So
$m_L = 20$
$m_R = 4.$
$m_L^+ = 8$
$m_R^+ = 4.$

Now $p_L^+ = \frac{m_L^+}{m_L} = \frac{8}{20}$

So gini_index(left_side) $N_L = p_L^+(1 - p_L^+) = 0.24$

Similarly $p_R^+ = \frac{m_R^+}{m_R} = \frac{4}{4}$

So gini_index(right_side) $N_R = p_R^+(1 - p_R^+) = 0$

Now, total gini_index $N = \frac{m_L}{m} N_L + \frac{m_R}{m} N_R = 0.24 * \frac{20}{24} + 0 * \frac{4}{24} = 0.2$

Therefore change in gini_index = 0.05

Now calculating the entropies for split points along x_2.

Consider the partition made by checking $x_2 <= \frac{1}{4}$.

So
$m_L = 8$
$m_R = 16.$
$m_L^+ = 3$
$m_R^+ = 9.$

Now $p_L^+ = \frac{m_L^+}{m_L} = \frac{3}{8}$

So gini_index(left_side) $N_L = p_L^+(1 - p_L^+) = 0.2344$

Similarly $p_R^+ = \frac{m_R^+}{m_R} = \frac{9}{16}$

So gini_index(right_side) $N_R = p_R^+(1 - p_R^+) = 0.2461$

Now, total gini_index $N = \frac{m_L}{m} N_L + \frac{m_R}{m} N_R = 0.2344 * \frac{8}{24} + 0.2461 * \frac{16}{24} = 0.2422$

Therefore change in gini_index = 0.0078

Now consider the partition made by checking $x_2 <= \frac{1}{2}$.

So
$m_L = 12$
$m_R = 12.$
$m_L^+ = 4$
$m_R^+ = 8.$

Now $p_L^+ = \frac{m_L^+}{m_L} = \frac{4}{12}$

So gini_index(left_side) $N_L = p_L^+(1 - p_L^+) = 0.2222$

Similarly $p_R^+ = \frac{m_R^+}{m_R} = \frac{8}{12}$

So gini_index(right_side) $N_R = p_R^+(1 - p_R^+) = 0.2222$

Now, total gini_index $N = \frac{m_L}{m} N_L + \frac{m_R}{m} N_R = 0.2222 * \frac{12}{24} + 0.2222 * \frac{12}{24} = 0.2222$

Therefore change in gini_index = 0.0278.

Finally considering the partition made by checking $x_2 <= \frac{3}{4}$.

So
$m_L = 8$
$m_R = 16.$
$m_L^+ = 5$
$m_R^+ = 7.$

Now $p_L^+ = \frac{m_L^+}{m_L} = \frac{5}{8}$

So gini_index(left_side) $N_L = p_L^+(1 - p_L^+) = 0.2344$

Similarly $p_R^+ = \frac{m_R^+}{m_R} = \frac{7}{16}$

So gini_index(right_side) $N_R = p_R^+(1 - p_R^+) = 0.2461$

Now, total gini_index N = $\frac{m_L}{m} N_L + \frac{m_R}{m} N_R = 0.2344 * \frac{8}{24} + 0.2461 * \frac{16}{24} = 0.2422$

Therefore change in gini_index = 0.0078.

Since, change in gini_index is maximum at the split point $\frac{1}{2}$ along x_1,
The root node should be $x_1 <= \frac{1}{2}$

5 Solution: Problem 5

a) Refer to attached code.
b) Refer to attached code.
c) The following table shows classification error (as calculated by classification_error.m) on the training set.

Train data classification error

Kernel	C = 0.01	C = 1	C = 100
Linear	0.155	0.154	0.154
Poly (deg = 2)	0.154	0.154	0.154
Poly (deg = 3)	0.151	0.153	0.153
RBF ($\sigma^2 = 1$)	0.115	0.099	0.052
RBF ($\sigma^2 = 4$)	0.167	0.148	0.126

The following table shows classification error (as calculated by classification_error.m) on the test set.

Test data classification error

Kernel	C = 0.01	C = 1	C = 100
Linear	0.165	0.166	0.166
Poly (deg = 2)	0.168	0.167	0.167
Poly (deg = 3)	0.166	0.165	0.165
RBF ($\sigma^2 = 1$)	0.129	0.127	0.197
RBF ($\sigma^2 = 4$)	0.186	0.170	0.143

As is obvious from the above data, the best classifiers for each kernel were

i) Linear kernel with C = 0.01.
ii) Polynomial kernel of degree = 3 with C = 1.
iii RBF kernel with C = 1 and $\sigma^2 = 1$.

The decision boundaries for these configurations follow:

Decision Boundary for Linear Kernel with C = 0.01

Decision Boundary for Polynomial Kernel of degree = 3 with C = 1

Decision Boundary for RBF Kernel with C = 0.01 and $\sigma^2 = 1$

Note: The above boundaries were made with resolution = 0.3. This was done to get the results a bit faster.

From the results obtained, it seems like there's no golden bullet which can aid in determining the optimal parameters in a real world situation. However, we can determine relatively good choices, by using techniques like cross validation.

d) Learning Curves for Handwritten Digit Classification using SVMs (with RBF kernel whose $\sigma^2 = 1$)

./HW1/Problem_5/dec_digit_classification.jpg

Clearly, these results are better than those of 5NN. This suggests that SVMs are better at classification than k Nearest Neighbours (though my results seems to be too good to be true, may be some implementation error).

6 Solution: Problem 6

This can be done by modifying the geometric margins. Intuitively, this can be done by associating " negative weights" to margins, where these weights will be a function of the distance as well as the error cost associated. So, the SVM algorithm will then try to maximize the weighted margins, thereby giving a classifier where different types of errors are treated differently.