Concordia University

Computer Science and Software Engineering Department

COMP 6321

Machine Learning

Assignment: 3

Name: Deep Pandya

Student ID: 27162391

Question 1:

(a)

Algorithm: Logistic Regression

Reason: In the given learning problem, it has few features involved and comparatively large training examples. So Logistic Regression algorithm would work perfectly fine.

(b)

Algorithm: Decision Tree

Reason: The classifier has to be justified to the board of education, it must be easy to explain and exhibit. So a Decision Tree would work well. Because it can be easily visualized and make it simple to understand what is happening under the hood.

(c)

Algorithm: Perceptron method

Reason: The requirement is the classifier has to be updated frequently when there are new data collected, thus we need an online learning method.

All items, which represent whether or not one specific custom goes for a product, is either 0 or 1. Thus it's a binary classification and can be processed with one sample each time, Perceptron will be rather efficient here.

(d)

Algorithm: Naïve Bayesian classifier

Reason: There is large attributes, and by comparison, rather small training samples involved. And some attributes are noisy. It's good to use Naïve Bayesian classifier, because first it's robust to noise, and then it works well using small training samples.

Q:-2 (a)

⇒ We have,

$P(0,0) = 1/3$, $P(0,1) = 1/3$, $P(1,0) = 0$,
$P(1,1) = 1/3$

The margin probability can be calculated as,

$P(x=0) = 2/3$, $P(x=1) = 1/3$,
$P(y=0) = 1/3$, $P(y=1) = 2/3$.

(i)

$$H[x] = \sum_i P_i \log \frac{1}{P_i}$$
$$= P_{x=0} \log \frac{1}{P_{x=0}} + P_{x=1} \log \frac{1}{P_{x=1}}$$
$$= \frac{2}{3} \log \frac{3}{2} + \frac{1}{3} \log 3$$
$$= 0.64$$

(ii)

$$H[y] = \sum_i P_i \log \frac{1}{P_i}$$
$$= P_{y=0} \log \frac{1}{P_{y=0}} + P_{y=1} \log \frac{1}{P_{y=1}}$$
$$= \frac{1}{3} \log 3 + \frac{2}{3} \log \frac{3}{2}$$
$$= 0.64$$

(iii)

$$H[Y|X] = P(X=0)\,H[Y|X=0] + P(X=1)\,H[Y|X=1]$$

Two conditional entropies can be expressed as,

$$H[Y|X=0] = P(Y=0|X=0)\log\frac{1}{P(Y=0|X=0)} + P(Y=1|X=0)\log\frac{1}{P(Y=1|X=0)}$$

$$H[Y|X=1] = P(Y=0|X=1)\log\frac{1}{P(Y=0|X=1)} + P(Y=1|X=1)\log\frac{1}{P(Y=1|X=1)}$$

According to the question, we have,

$P(Y=0|X=0) = 1/2$ $\quad$ $P(Y=1|X=0) = 1/2$

$P(Y=0|X=1) = 0$ $\quad$ $P(Y=1|X=1) = 1$

So,

$$H[Y|X] = \frac{2}{3}\left[\frac{1}{2}\log 2 + \frac{1}{2}\log 2\right] + \frac{1}{3}[0+0]$$

$$= \frac{2}{3}\log 2$$

$$= 0.46$$

(iv)

$$H[x|y] = P(y=0)\,H[x|y=0] + P(y=1)\,H[x|y=1]$$

$$= P(y=0)\left[P(x=0|y=0)\log\frac{1}{P(x=0|y=0)} + P(x=1|y=0)\log\frac{1}{P(x=1|y=0)}\right]$$

$$+ P(y=1)\left[P(x=0|y=1)\log\frac{1}{P(x=0|y=1)} + P(x=1|y=1)\log\frac{1}{P(x=1|y=1)}\right]$$

and we have the conditional probability,

$P(x=0|y=0) = 1$ $\quad$ $P(x=0|y=1) = 1/2$

$P(x=1|y=0) = 0$ $\quad$ $P(x=1|y=1) = 1/2$

so,

$$H[x/y] = \frac{1}{3}[0+0] + \frac{2}{3}\left[\frac{1}{2}\log 2 + \frac{1}{2}\log 2\right]$$

$$= \frac{2}{3}\log 2$$

$$= 0.46$$

(v)

$$H[x,y] = \sum_i P_i(x,y) \log \frac{1}{P_i(x,y)}$$

$$= \frac{1}{3}\log 3 + \frac{1}{3}\log 3 + 0 + \frac{1}{3}\log 3$$

$$= \log 3$$

$$= 1.099$$

(vi)

$$I[x,y] = I[x/y]$$

$$= H[x] - H[x/y]$$

$$= \frac{2}{3}\log\frac{3}{2} + \frac{1}{3}\log 3 - \frac{2}{3}\log 2$$

$$= \frac{1}{3}\log\frac{27}{16}$$

$$= 0.17$$

Q:-2 (b)

for function, $f(p_i) = \ln \frac{1}{p_i} = -\ln p_i$,

$$\frac{\partial^2 f(p_i)}{\partial p_i^2} = \frac{1}{p_i^2} \geq 0,$$

Thus, f is a convex function, and according to jensen's inequality,

$$f\left(\sum_{i=1}^{M} \lambda_i x_i\right) \leq \sum_{i=1}^{M} \lambda_i f(x_i),$$

where $\lambda_i \geq 0$ and $\sum_{i=1}^{M} \lambda_i = 1$

In this case we can have $p_i = \lambda_i$, $f(x_i) = -\ln(x_i)$ and $x_i = \frac{1}{p_i}$,

it satisfied Jensen's inequality as proved above, then we can have,

$$-H[x] = -\sum_{i=1}^{M} p_i \ln \frac{1}{p_i} \geq -\ln\left[\sum_{i=1}^{M} p_i \frac{1}{p_i}\right]$$

$$= -\ln M,$$

which equals to,

$$H[x] \leq \ln M$$

This is to say that the largest entropy of x is $\ln M$.

In the case of uniform distribution, where we have $P_i = \frac{1}{M}$, the entropy is,

$$H[x] = \sum_{i=1}^{M} P_i \ln \frac{1}{P_i}$$
$$= \ln \frac{1}{P_i}$$
$$= \ln M$$

So far, we get the conclusion that when P is a uniform distribution, it can get the highest entropy.

Thus, the question is proved.

(c)

$\Rightarrow$ The information gain formula can be derived as following.

$$IG(D|Test) = H[D] - H[D|Test]$$
$$= H[D] - P[Test = t] + P[Test = f)$$
$$H[D|Test = f]$$

for the two test, $H[D]$ remains the same, so we only need to compare $H[D|Test]$.

It can be told from the formula that the test which generates smaller $H[D|Test]$ wins.

For T_1, we have,

$$P_1(Test = t) = \frac{30}{40} = \frac{3}{4}$$

$$P_1(Test = f) = \frac{10}{40} = \frac{1}{4}$$

$$H_1[D|Test = t] = \frac{20}{30} \log \frac{30}{20} + \frac{10}{30} \log \frac{30}{10}$$

$$= \frac{2}{3} \log \frac{3}{2} + \frac{1}{3} \log 3$$

$$H_1[D|Test = f] = 0$$

Thus, its Conditional entropy of H[D|Test] is,

$$H[D|Test] = \frac{3}{4}\left[\frac{2}{3}\log\frac{3}{2} + \frac{1}{3}\log 3\right] + 0$$
$$= \frac{1}{4}\log\frac{27}{4}$$
$$= 0.48$$

For T_2,

$$P_2(Test = t) = \frac{22}{40} = \frac{11}{20}$$

$$P_2(Test = f) = \frac{18}{40} = \frac{9}{20}$$

$$H_2[D|Test = t] = \frac{15}{22}\log\frac{22}{15} + \frac{7}{22}\log\frac{7}{22}$$

$$H_2[D|Test = f] = \frac{15}{18}\log\frac{18}{15} + \frac{3}{18}\log\frac{18}{3}$$
$$= \frac{5}{6}\log\frac{6}{5} + \frac{1}{6}\log 6$$

Thus, its conditional entropy of H[D|Test] is,

$$H_2[D|Test] = \frac{11}{20}\left[\frac{15}{22}\log\frac{22}{15} + \frac{7}{22}\log\frac{22}{7}\right] + \frac{9}{20}\left[\frac{5}{6}\log\frac{6}{5} + \frac{1}{6}\log 6\right]$$

$$= \frac{3}{8}\log\frac{22}{15} + \frac{7}{40}\log\frac{22}{7} + \frac{3}{8}\log\frac{6}{5} + \frac{3}{40}\log 6$$

$$= 0.547$$

So far, we have calculated that

$H_2[D|Test] > H_1[D|Test]$; thus $IG_1(D|Test) > IG_2(D|Test)$; that test 1 wins.

Q:-3

⇒ (a), (c) and (e) are valid kernel, (b) and (d) are not.

⇒ Reasons :-

A function is a kernel as long as it's symmetric and positive semi-definite and vice-versa.

According to the question,

$k_1(x,z)$ and $k_2(x,z)$ are kernels, thus, we can have,

$$k_1(x,z) = k_1(z,x) \geq 0$$

$$k_2(x,z) = k_2(z,x) \geq 0$$

(a) $$k(z,x) = a k_1(z,x) + b k_2(z,x)$$
$$= a k_1(x,z) + b k_2(x,z)$$
$$= k(x,z) \geq 0$$

So, $k(x,z) = k_1(x,z) + k_2(x,z)$ satisfies Mercer's theorem, is a valid kernel.

(b) $k(x,z) = ak_1(x,z) - bk_2(x,z)$, $a,b \geq 0$

thus, $k(x,z)$ is not necessarily always positive, it does not satisfy the positive semi-definite condition.

so, (b) is not a valid kernel.

(c)

$$k(x,z) = k_1(x,z)\, k_2(x,z)$$
$$= k_1(z,x)\, k_2(z,x)$$
$$= k(z,x) \geq 0$$

and it meets both condition, and thus, (c) is a valid kernel.

(d)

Since $f: R^n \to R$ is a mapping function from n dimension to 1 dimension, we can have:

$$k(x,z) = f(x)\, f(z) = f(z)\, f(x) = k(z,x)$$

meaning that k satisfies the condition of being symmetric, and for second condition of being positive semi-definite, consider an arbitrary vector V, we have:

$$V^T k(x,z) V = V^2 f(x) f(z)$$

Since there is no restriction put on value of $f(x)$, there is a chance that $f(x) f(z) < 0$, thus it doesn't prove the second claim, and is not a valid kernel.

(e)

since p is a probability density function, it can be seen as a mapping from higher dimension to one dimension, as is f in the question (d).

The difference is that it's guaranteed that $P \geq 0$. So for ~~many~~ any arbitrary vector V, we have:

$$V^T k(x,z) V = V^2 P(x) P(z)$$

Because $V^2 \geq 0$ and $P(x) \geq 0$, $P(z) \geq 0$ thus (e) satisfies both condition of being symmetric as well as positive semi-definite, and is a valid kernel.

Q 3-4

⇒ They are different.

Reason :-

for a 2 - attribute case, the 1-nearest neighbour method employs a Voronoi diagram, where in a space all samples are distributed and for each sample, the space is divided in such a way that points consisting the corresponding region are closer to that sample than to any other.

However, in the method of decision tree, all the boundaries are drawn in a way that it's either vertical or horizonted, because in each step, the distance will be compared in terms of only one attribute.

Since, there existing two different ways to split the space for same dataset, for sure that the result will diverge for some sample, thus it's proved the two above mentioned method are different.

Q:-5 (a)

⇒ We have,

$$P(\omega_i) = 1/c$$

Error rate upper bound will be,

$$P \leq P^* \left(2 - \frac{C}{C-1} * P^*\right)$$

Because, each category probability is same, than for all x, $P(x/\omega_i)$ is same for, $i = 1, 2, 3, \ldots, C$

Each category has some probability

$$P(\omega) = 1/C, \quad C = 1, \ldots, C$$

According to Bayes,

$$P(\omega_i/x) = \frac{P(x/\omega_i)\ P(\omega_i)}{P(x)}$$

Posterior probability $P(\omega_i/x)$ will be the same $i = 1, \ldots, C$

Thus $P(\omega_i/x) = 1/C$

(b)

Because posterior probability $P(\omega_i|x)$ will be same, than according to Bayes decision ; we can distinguish x as any of a category ω_m. and Because of

$P(\omega_m|x) = 1/c$,

the error rate for different x is:

$$P(e|x) = 1 - 1/c$$

$$\therefore p^* = \int P(e|x) \cdot P(x)\, dx$$

$$= 1 - 1/c$$

Now calculate nearest neighbour error rate P,

$$P_n(e|x) = \int \left[1 - \sum_{i=1}^{c} P(\omega_i|x)\, P(\omega_i|x')\right] \delta(x'-x)\, dx'$$

$$= 1 - \sum_{i=1}^{c} p^2(\omega_i|x)$$

$$\therefore P = \int \left[1 - c * \frac{1}{c^2}\right) P(x)\, dx = 1 - \frac{1}{c}$$

∴ we make a conclusion that the nearest neighbour rate is $P = p^*$

Question 6

There are two files involved. They are *nFolderKNearestNeighbour.m, and validation.m.*

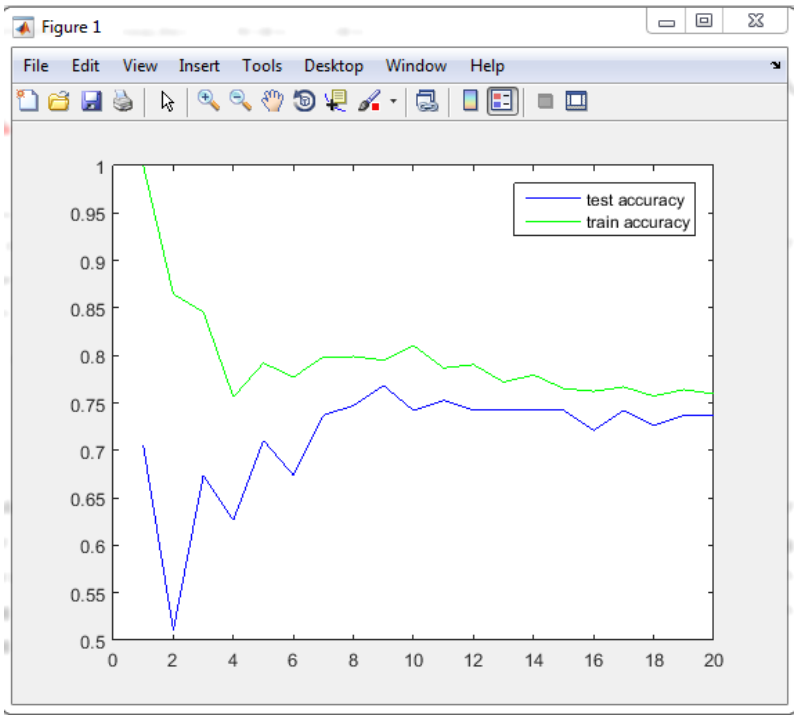
In *KNearestNeighbour.m,* there are four arguments: dataset, Y, n, and k. Dataset is the input of features, Y is the class result, user can set how many fold they use as n, and how many nearest neighbors to employ as k. The return values are testAccuracy and trainAccuracy that respectively stand for the testing and training accuracy for a particular number of fold n and nearest neighbor k.

To call the function, user need to specify what is the input of features and supervised results, as well as how many folds and nearest neighbors they want to employ. In this question, feature matrix is wpbcx, result is wpbcy, and by convention, we set n to 10 folds, and let's assume we use 5 nearest neighbors (but the values can be set by the user, they don't have to be the same). Thus, in the command window, it would be like this:

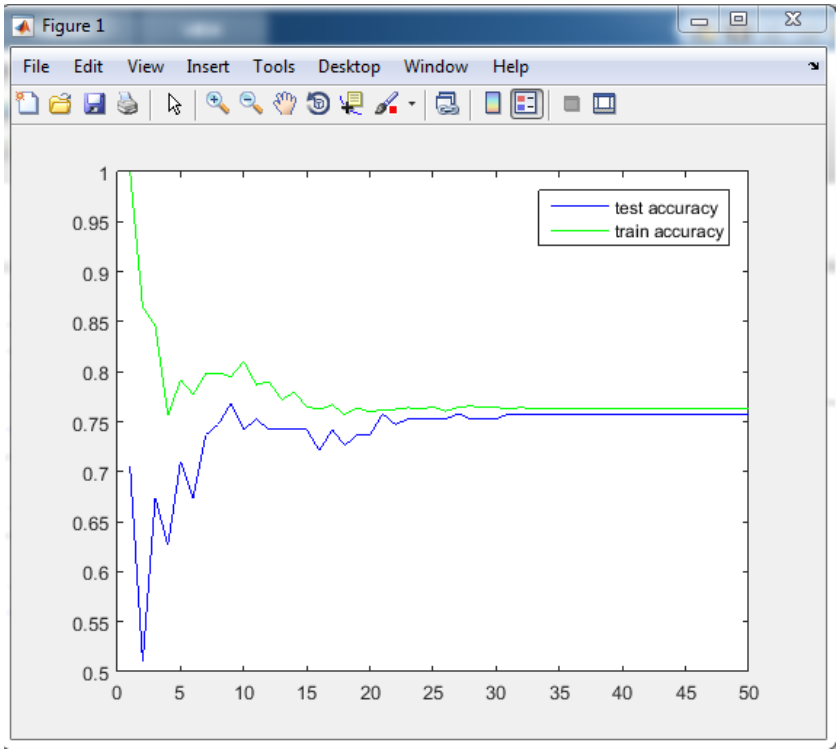
nFolderKNearestNeighbour(x,y, 10, 1)

In *validation.m, the arguments stand for same meaning as that in nFolderKNearestNeighbour.m. However, the output* testAccuracy and trainAccuracy are now vectors that their values stand for the average accuracy of k from 1 to the maximum. For example, if the user wants to see how the accuracy changes as k goes from 1 to 20, they can write in the command line (assume we still use 10 folds):

Here is how the result look like:



And following is another figure showing how the accuracy change as number of nearest neighbor goes from 1 to 51. It should be noted that the highest testing accuracy takes place when k equals to 9. And when k get large, the accuracy of both training and testing set will converge. Also, please note that in calculating both training and testing accuracy, I use the same training dataset.



To validate more results, you can set whatever is the largest number of nearest neighbors by calling the ***validation*** function.

References:

- Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006
- Kevin P. Murphy, *Machine Learning. A Probabilistic Perspective,* MIT Press, 2012.
- http://www.rmki.kfki.hu/~banmi/elte/Bishop%20-%20Pattern%20Recognition%20and%20Machine%20Learning.pdf
- Jensen's Inequality (Pattern Recognition and Machine Learning, Bishop, Page 56) for question 2.
- Help from friend who is doing major in mathematics at McGill University and classmates.