Homework 2: HMMs and Machine Learning

Due date: Nov/13

1) (40 pts) Consider a Markov Model with a binary state X (i.e., x_t is either 0 or 1). The transition probabilities are given as follows:

X_t	X_{t+1}	$P(X_{t+1} \mid X_t)$
0	0	0.85
0	1	0.15
1	0	0.4
1	1	0.6

a. Draw the Markov chain (state machine) capturing this variable's transition probabilities.

b. The prior belief distribution over the initial state X_0 is 0, i.e., $P(X_0 = 0) = 1$ and $P(X_0 = 1) = 0$. What is the belief distribution after three steps: $P(X_1)$, $P(X_2)$ and $P(X_3)$?

Now, we incorporate sensor readings. The sensor model is parameterized by a number $k \in [0,1]$

X_t	E_t	$P(E_t \mid X_t)$
0	0	k
0	1	$(1 - k)$
1	0	$(1 - k)$
1	1	k

c. At t = 3, we get the first sensor reading, E3 = 1. Use your answer from part (b) to compute $P(X_3 \mid E_3 = 1)$. Leave your answer in terms of k.

d. For what range of values of k will a sensor reading E_3=1 increase our belief that $X_3 = 0$? In other words, what is the range of k for which $P(X_3 = 0 \mid E_3 = 1) > P(X_3 = 0)$? Defend

e. Unfortunately, the sensor breaks after just one reading, and we receive no further sensor information. Compute $P(X_\infty \mid E_3 = 1)$, the stationary distribution very many time steps from now.

f. How would your answer to part (e) change if we never received the sensor reading E3, i.e. what is $P(X_\infty)$ given no sensor information?

Hints: The basic HMM update equations are:
Passage of time:

$$P(X_{t+1}) = \sum_{x_t}[P(X_{t+1}|x_t)P(x_t)],\text{ where } X_t \text{ is either } \{x_t = 0, x_t = 1\}$$

Observation (sensor reading):

$$P(X_t|e_t) = \frac{P(e_t|X_t)P(X_t)}{\sum_{x_t}[P(e_t|X_t)P(X_t)]}$$

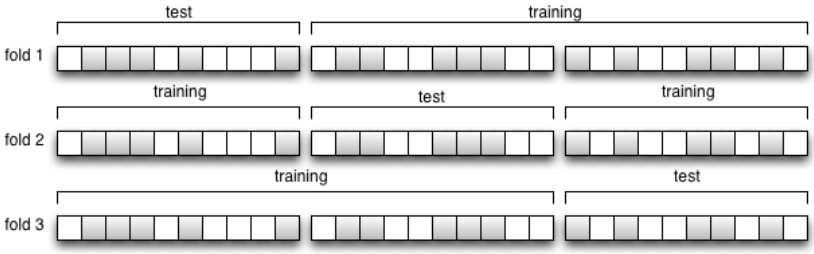
To calculate the stationary distribution you need to make the prior (previous time step) equal to the posterior (next time step) in the equation for passage of time, and also consider that the probability of $P(X_\infty=0)$ and $P(X_\infty=1)$ must sum to one.

2) (10 pts) In this problem you will explore and analyze the Pima dataset (download from mycourses). The dataset consists of 8 attributes and a binary attribute defining the class label, the presence of diabetes. Data entries are organized in rows such that attributes come first and the class label is last. Before applying learning algorithms some data preprocessing may be necessary.

a. Write a function *normalize* that takes an unnormalized vector of attribute values and returns the vector of values normalized according to the data mean and standard deviation. The normalized value should be: $x_{norm} = \frac{x-\mu_x}{\sigma_x}$, where x is an unnormalized value, μ_x is the mean value of the attribute in the data and σ_x its standard deviation. Test your function on attribute 3 of the pima dataset. Normalize all inputs of the Pima dataset. There is no need to normalize the output (last column).
Hint: You may use the matlab functions `importdata`, `mean` and `std`. To verify that your normalization is correct, you may plot a visualization of the variance of the data before and after normalization using the function `boxplot`.

b. Write a function *divideset* that randomly splits the dataset into three non-overlapping datasets to perform K-fold cross-validation. The algorithm should always give you different training and test sets (random sampling). Divide the Pima dataset into three sets of approximately equal size (about 33% of the data each).
Hint: You may use the matlab function `crossvalind`.

Note: Cross-validation is used to evaluate or compare algorithms as follows: in each iteration, the machine learning algorithms use k-1 folds of data to train models, and subsequently the trained models are asked to make predictions about the data in the test fold. The performance of each algorithm can be calculated using a metric such as accuracy. To obtain an aggregate measure from the k folds, the mean can be calculated over all folds. Alternatively, the metrics can be calculated after each fold's results are recombined as a single output vector. The procedure of 3-fold cross-validation is illustrated below.



3) (50 pts) Use your normalized Pima dataset to compare the performance of the following machine learning algorithms using K-fold cross-validation. You will have to extract the last column of the Pima dataset as Y labels. You should experiment with the parameters of each learning algorithm.

i) Boosting (decision trees)
Parameter to test: number of learners 'NLearn'.

ii) Neural network (multilayer feed-forward)
Parameters to test: number of nodes in the hidden layer 'hiddenSizes'.

iii) Support vector machine (Gaussian radial basis function kernel)
Parameters to test: the RBF kernel parameter 'rbf_sigma'; you can also test other kernel functions 'kernel_function', such as polynomial.

Describe your results responding to the following questions.

a. Report the cross-validation results (combining the tree test set folds) for each method above. You should calculate **confusion matrices, accuracy, precision and recall**.

b. For each method, what parameters provided the best results?

c. According to your results, what was the best performing algorithm overall?

d. What was the computation time of the winning algorithm compared to the others?

e. Based on the theory underlying each machine learning method, explain why you think the algorithms behaved the way they did. [This is the most important question. It is worth double points]

Hint: Use the example code provided. You may use the matlab functions `confusion` or `confusionmat`, or even `classperf.` You can verify your answers using matlab's new 'Classification Learner App', but you should implement your own functions to solve problems (2) and (3).

What to hand in

Upload to myCourses' dropbox two separate files: your write-up and responses in PDF, and your Matlab scripts (zipped if more than one file). A "write-up" is a brief report explaining how you did it and your observations (analysis of results, any issues, etc). You must include your code to support your write-up. However, the write-up should be a self-contained document. In other words, I will assess your homework based on your write-up alone. I will run your code only to confirm your results if needed.

Show all your calculation and justify your answers. You can scan your handwritten response (or take a photo with your cellphone) instead of typing in the computer. Include your name as part of the filename.