Submission until: **05.07.2015**
Discussion on: 07.07.2015
Submission as upload to your groups stud.IP folder as *groupNumber_sheet10.zip*

Assignment 1 *(Probabilities (2p))*

Suppose we have three bags containing colored chocolate candy. These bags($b1, b2, b3$) contain the amounts given in Table 1 below.

	red candy	green candy	yellow candy
$b1$	5	3	2
$b2$	1	2	3
$b3$	4	2	5

1. Let's assume we grab one candy from bag $b1$ at random (no see through bag). What is the probability of getting a red one/ green one/ yellow one?
2. If one bag is chosen at random with probabilities $p(b1) = 0.2$, $p(b2) = 0.3$, $p(b3) = 0.5$, and from that bag we grab one candy at random. Then what is the probability of getting a red one/green one/yellow one?

Assignment 2 *(Bayes classifier (8p))*

In the following, we consider a data set where the task is to describe whether a person is ill. We use a representation based on three features per subject to describe an individual person. These features are *running nose*, *coughing*, and *reddened skin*, each of which can take the value true (+) or false (−), see Table 2.

Training Example	N (running nose)	C (coughing)	R (reddened skin)	F (fever)	Classification
d1	+	+	+	-	positive (ill)
d2	+	+	-	-	positive (ill)
d3	-	-	+	+	positive (ill)
d4	+	-	-	-	negative (healthy)
d5	-	-	-	-	negative (healthy)
d6	-	+	+	-	negative (healthy

1. Given the data set in Table 2, determine all probabilities required to apply the Bayes classifier for predicting whether a new person is ill or not. Use a naive approach, where all attributes are assumed to be independent. (4p)
2. Calculate the probabilities of persons d1,.., d6 being ill. (2p)
3. Calculate the probabilities of the following subjects being ill: (2p)
   a) a person who is coughing and has fever
   b) a person whose nose is running and who suffers from fever
   c) a person with a running nose and reddened skin

Assignment 3 *(Reinforcement Learning (10p))*

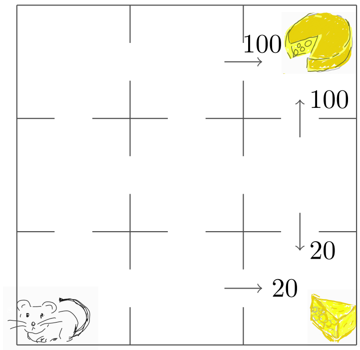

Figure 1: A mouse and cheeses. Possible transitions are up ↑, left ←, right →, down ↓. The position of the biggest cheese (top right) is our goal. Values next to arrows indicate immediate reward. No value indicates an immediate reward of 0.

1. Approximating the Value Function
   a) The discounted cumulative reward $V(s_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ... = \sum_{t'=t..\infty} \gamma^{t'} r_{t'}$
   b) Assume the action sequence of the mouse (→↑←↑→→) is applied to the Figure 1. Calculate $V(s_t)$ with discount factor $\gamma = 0.9$. (2p)
2. Q-Learning
   a) Now, we will apply Q-learning to the Figure 1. The reward is zero almost everywhere, except when entering the state where the cheeses located. The training consists of a series of episodes. During each episode, the agent begins at some randomly chosen state and is allowed to execute actions until it reaches the goal (the biggest cheese). When the agent(mouse) reaches the goal state, the episode ends and the agent is moved to a new, randomly chosen initial state for the next episode. Q-Learning start with all Q values initialized to 0.

   b) Let's start the first episode from the lower left corner where the mouse is. Perform three episodes of Q-Learning for this example. (8p)

Assignment 4 *(Classification (10p))*

1. Naive Bayes Classification (5p)
   a) Load MNIST dataset *mnist_train_09.mat* and compute the priori probability for each class
   b) Use a buit-in function for Naive Bayes approach in Matlab or *sklearn* package for Python to predict *mnist_test_09.mat*. Save five figures from this test dataset that failed- and succeeded classified each.
2. Support Vector Machine (SVM) (5p)
   a) SVM is an inherently two-class classifier. Some of SVM libraries, e.g. buit-in function in Matlab, do not provide function for multi-class data. If we want to use SVM to solve multi-class classification problem such as MNIST, shortly explain how can we solve this problem?

b) Load *data_train.mat* which contains data in two classes. Plot these data with different color corresponding to target output. Select an appropriate kernel, then use the built-in SVM model to train the data. Compute the prediction of *data_test.mat* and plot the predicted output on the same figure.

c) **(Optional)** You can apply SVM with MNIST dataset. For Matlab, you may apply the method from a) or use external library for multi-class SVM to solve this multi-class problem or . (0p)

Assignment 5 *(LDA (6p))*

a) Discuss the purpose of LDA in comparison with PCA.

b) Compute LDA projection for the following 2D dataset which is discussed in the lecture (p.11)
class C1 has dataset = $\{(4,1),(2,4),(2,3),(3,6),(4,4)\}$
class C2 has dataset = $\{(9,10),(6,8),(9,5),(8,7),(10,8)\}$
Algorithm:

1) Formulate the data
2) Compute mean for each class, μ_1 and μ_2
3) We formulate criteria for class separability from so called with-in class scatter.
Within-class scatter, S_w, is the expected covariance of each classes. For two-class, we can compute $S_w = p_1 \times cov_1 + 0.5 \times cov_2$.
The covariance matrix is computed from: $cov_j = (x_j - \mu_j)(x_j - \mu_j)^T$.
4) Therefore, the optimizing criterion for the class independent transform is computed as: $\overrightarrow{w} = {S_w}^{-1} \times (\mu_1 - \mu_2)$

c) Sketch the data and $\overrightarrow{w}$