Examination

Linköping University, Department of Computer and Information Science, Statistics

Course code and name	732A99/732A68 Machine Learning
Date and time	2021-08-25, 8.00-13.00
Assisting teacher	Oleg Sysoev
Allowed aids	See "732A99_TDDE01_exam_regulations.PDF"
Grades:	A=19-20 points plus passed oral defense
	B=16-18 points plus passed oral defense
	F= 16-20 points plus failed oral defense
	C= 16-20 points without oral defense
	C=11-15 points with or without oral defense
	D=9-10 points with or without oral defense
	E=7-8 points with or without oral defense
	F=0-6 points with or without oral defense

Provide a detailed report that includes plots, conclusions and interpretations. Give motivated answers to the questions. If an answer is not motivated, the points are reduced. Provide all necessary codes in the appendix.

Use seed 12345 when randomness is present unless specified otherwise.

Assignment 1 (10p)

Part 1

1. Implement a function that for given numbers n and p generates a data set with n observations and p features and one target where each observation (row) is generated according to the following:

$$X_i \sim U[0,1], i = 1, \dots, p$$

$$Y = \begin{cases} 1 \; if \; \sum_{i=1}^{p} X_i < 0.5p + \epsilon, & where \; \epsilon \sim N(0,0.1) \\ 0, otherwise \end{cases}$$

By using this function, implement a loop for $p = 2, \dots, 100$ where you generate training data with 100 observations, test data with 200 observations (specify set.seed before the loop) and then fit a 3-nearest neighbor classifier to these data in the loop. Present a dependence of the test accuracy on p as a scatter plot and comment on the trend you observe and which phenomenon this trend demonstrates. **(3p)**

2. Use same function as in step 1 to simulate training data with 100 points and test data with 1000 points and dimension $p = 3$ (use set.seed once before simulating the training data) and then fit a K-nearest neighbor classifier to these data where $K = 1, \dots, 99$. Provide a plot showing dependence of the test accuracy on K. What kind of trend can you see in this plot and what phenomenon is demonstrated by this trend? **(2p)**

Part 2

Data file **women.csv** contains clinical records about female cancer patients and variable Death that represent the survival status (0=survived, 1=not survived)

1. Divide these data into training and test data (50/50) and compute a classification tree which predicts Death variable based on the remaining variables; do this by first growing the full tree with minimum deviance parameter 0.003 and then computing the optimal tree by the cross-validation. Report a) the plot showing the dependence of the cross-validation error on the tree size b) how many features are selected by the tree c) confusion matrix for the test data and d) misclassification error. **(3p)**
2. Use the same data partitioning to fit a generalized additive model with features Blood Systolic and Cholesterol and response Death so that the model has a correct number of knots corresponding to the smoothing splines models and compute the confusion matrix for the test data and the test misclassification error. Which of these two models – GAM or tree - would you prioritize here and why? What kind of problem is demonstrated by these data when computing both models? **(2p)**

Assignment 2 (10p)

Part 1: Mixture models (4p)

You are asked to implement the EM algorithm for mixtures of multivariate Gaussian distributions. You should use the following equations in the E-step

$$p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \frac{\pi_k f(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_k \pi_k f(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

and in the M-step

$$\begin{aligned}
\pi_k^{ML} &= \frac{\sum_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})}{N} \\
\boldsymbol{\mu}_k^{ML} &= \frac{\sum_n \boldsymbol{x}_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})}{\sum_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})} \\
\boldsymbol{\Sigma}_k^{ML} &= \frac{\sum_n (\boldsymbol{x}_n - \boldsymbol{\mu}_k^{ML})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{ML})^T p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})}{\sum_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})}
\end{aligned}$$

where f is the density function of a multivariate Gaussian distribution, which is implemented by the function `dmvnorm` in the R package `mvtnorm`. The learning data (600 2-D points) can be obtained via `read.table("dataEM.txt")`.

(2 p) Implement the EM algorithm as described above. Use the following initial parameter estimates: $\boldsymbol{\mu}_k = (u, v)$ where u and v are uniformly chosen in the interval [0,5], and $\boldsymbol{\Sigma}_k = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ for all k. Run your code with three components on the data provided. Show that the log likelihood increases with the number of iterations. Show also the final parameter estimates.

(2 p) The model learned by the EM algorithm above is unrestricted in the sense that it does not assume any independence among the random variables i.e. the off-diagonal elements of $\boldsymbol{\Sigma}_k$ may be non-zero for all k. Alternatively, we can assume a restricted model where all the random variables are conditionally independent of each other given the component, i.e. the off-diagonal elements of $\boldsymbol{\Sigma}_k$ are always zero for all k. Use the BIC to select between the restricted and unrestricted models. The BIC is defined as

$$LL - \frac{M}{2} \log N$$

where LL is the log likelihood of the learning data given the parameter estimates returned by the EM algorithm, M is the numbers of parameters in the mixture model at hand, and N is the number of points in the learning data. You may want to check the course slides or Bishop's book for more information on the BIC score.

Part 2: Kernel methods (6p)

In the slides 11 and 12 of the lecture on kernel methods, you can see how to produce a probabilistic classifier by using kernel density estimation and Bayes theorem. You are asked to implement such a classifier and estimate its generalization error. The learning data (2500 1-D points with their corresponding class labels) can be obtained via `read.table("dataKernel.txt")`. You should use the Gaussian kernel as implemented by the R function `dnorm`, i.e. the standard deviation in the function plays the role of kernel width h. Note that you want to estimate the generalization error while optimizing the hyperparameter h. One solution to this problem is to use 2×2 nested cross-validation:

1 Divide the learning data into approximately equal sized folds D_1 and D_2
2 Divide the fold D_1 into approximately equal sized folds D_{11} and D_{12}
3 For each hyperparameter value $h = 0.5, 1, 5, 10$ do
4 Train the classifier on D_{11} and validate it on D_{12}
5 Train the classifier on D_{12} and validate it on D_{11}
6 Compute the average error on the validation folds
7 Select the hyperparameter value with the lowest average validation error
8 Train the classifier on D_1 and test it on D_2
9 Repeat the steps 2-8 above swapping the roles of D_1 and D_2
10 Report the average error on the test folds

(4 p) Implement the pseudocode above.
(2 p) Select the classifier to return to the user, i.e. the value of the hyperparameter value h. Use any method that you deem appropriate.