

## Problem assignment 6

*Due: Wednesday, March 12, 2003*

In this homework we continue to study and experiment with "Pima" dataset. You can download the dataset (*pima.txt*) and its description (*pima\_desc.txt*) from the course web page. In addition to the complete dataset *pima.txt*, you have *pima\_train.txt* and *pima\_test.txt* you need to use for training and testing in problem 1.

### Problem 1. Support vector machines

Support vector machines represent yet another technique one can apply to the problem of binary classification. The idea is to find the hyperplane that separates the examples in two classes the best. The best hyperplane is defined in terms of the maximum margin. The learning problem reduces as usually to optimization, in this case, a quadratic optimization problem.

There is a number of implementations of SVM algorithms with better or worse running time performances. Here we use a Matlab code implementing SVM solver for the linear decision boundary proposed by O.L. Mangasarian and D. Musicant. The paper describing this method can be downloaded electronically at:

<http://www.ai.mit.edu/projects/jmlr/papers/volume1/mangasarian01a/html/>. The SVM solver is in files *svml.m* and *svml\_itsol.m* that can be downloaded from the course web page. *svml\_itsol.m* is a slightly modified version of the original program by O.L. Mangasarian and D. Musicant. To run it you call *svml.m* that takes care of converting outputs from 0,1 class labels to -1,1 and sets other parameters of the Lagrangian SVM.

Write and submit a Matlab program that:

- Loads training and test data.
- Calls linear SVM solver to learn the linear decision boundary;
- Computes the mean misclassification error for both the training and test data.
- Computes the confusion matrix for the test set.

In your report include the misclassification errors and confusion matrix obtained for the train and test sets. Compare the result to the results of the logistic regression model from the previous assignment (you can use the NN implementation given to you or your own code for this purpose).

**Optional credit.** If you are interested in SVMs with non-linear kernels and would like to try them on the pima dataset you can found and download the Matlab code for SVMs from <http://www.kernel-machines.org/> website. Try quadratic or cubic kernels.

## **Problem 2. Evaluating generalization performance**

The average test misclassification error measures the capability of a learner to classify correctly examples not used in the training (learning) stage and thus reflects its capability to generalize. However, by performing experiments with datasets we can observe that misclassification errors can vary depending on the test and training set split.

Write a program (you do not have to turn it in) that:

- Divides 'pima' data into the training and test set. About 30% of examples should go into the test set. Use function `divideset.m` you wrote earlier for this purpose.
- Repeats the experiment in problem 1 thirty (30) times, but each time on different training and testing data. Use `divideset.m` function to randomly divide the data (keep the 30-70% split fixed).
- Computes and reports the average and standard deviation of the mean misclassification results on both training and test datasets.

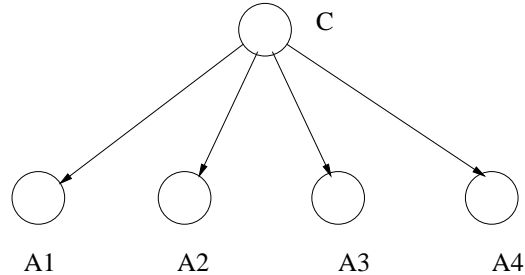
Report results obtained by averaging misclassification errors over different train/test splits. You should observe that the mean misclassification results vary for different training and testing choices. If you have to evaluate the learner and its capability to generalize on the learning problem, which would you prefer, averages or single train/test split? Explain why?

Use the same evaluation experiment for the logistic regression and neural network with 2 hidden nodes (from assignment 5). Which classifier do you think would be the best choice for the pima dataset?

## **Problem 3. Naive Bayes model**

A Naive Bayes model is a special type of a Bayesian belief network. It consists of a class variable  $C$  with values corresponding to different class types and feature variables (nodes) that characterize different class types. The model encodes probability distribution over all

variables with a very simple independence structure: features are independent given the class type. A Naive Bayes model with four features (A1-A4) is illustrated in the figure below.



Assume that all variables in the Naive Bayes model in the above figure are binary with two different values - True (T) and False (F).

Answer:

- (a) What is the number of parameters of the full joint distribution defined over all of the variables in the Naive Bayes model in the figure above?
- (b) Describe the parametrization of the Naive Bayes model. How many parameters are needed to define the model?
- (c) Show how to compute the joint probability  $P(C = T, A1 = T, A2 = T, A3 = F, A4 = T)$  using the parameters of the Naive Bayes networks.
- (d) Show how to compute the conditional probability  $P(C = T | A1 = T, A2 = T, A3 = F, A4 = T)$  from the parameters of the Naive Bayes network. Try to simplify the expression as much as possible.