Lehrstuhl für Informatik VI **Wednesday, November 23th, 2008**
Rheinisch–Westfälische Technische Hochschule Aachen **M. Ratajczak & S. Wiesler**
Prof. Dr.–Ing. H. Ney

6. Exercise Sheet **Pattern Recognition and Neural Networks**

The solutions to the problems indicated with (*...) may be submitted until **Wednesday, December 2nd, 2009**, either in the secretariat of the Lehrstuhl für Informatik VI or at the latest before the exercise lesson on the same day.
Please note: for the submission of programming problems we require that you use C/C++. Submit programs as printout, as well as by email as working code to both of us
`{wiesler,ratajczak}@i6.informatik.rwth-aachen.de` with the subject "[PatRec] Sheet 6".

1. Maximum Likelihood Estimation of Binomial Distributions

(a) Derive the Maximum Likelihood estimaton formula for the binomial distribution.

$$p_N(n|\vartheta) = \binom{N}{n} \vartheta^n (1-\vartheta)^{N-n}.$$

(b) Derive the Maximum Likelihood estimaton formula for the poisson distribution. (* 5P)

$$p(n|\lambda) = \frac{\lambda^n}{n!} \mathrm{e}^{-\lambda}.$$

2. Text Classification - Multinomial Distributions

In text classification, texts are grouped according to a set of topics, which are the classes for this problem. A basic approach to text classification is to use as observations the number of occurences of the set of words of a prespecified vocabulary. The class (topic) conditional probability distribution for a text would therefore be the probability to observe n_1-times word w_1, n_2-times word w_2, etc., which is modelled by the *multinomial* distribution:

$$p(n_1, n_2, \dots, n_r | N, \theta_1^r) = \frac{N!}{n_1! \cdot n_2! \cdot \dots \cdot n_r!} \cdot \theta_1^{n_1} \cdot \theta_2^{n_2} \cdot \dots \cdot \theta_r^{n_r},$$

with (class specific) parameters $\theta_1^r = \theta_1, \dots, \theta_r$, $\sum_{i=1}^{r} \theta_i = 1$, and with $N = \sum_{i=1}^{r} n_i$.

On our server wasserstoff.informatik.rwth-aachen.de (anonymous ftp) you find data for a text classification task in the directory pub/PatRec/textclassification/ with real data for a text classification problem based on a set of 20 newsgroups. The original corpus containing the corresponding messages posted into the corresponding newsgroups is contained in the file 20_newsgroups.tar.gz (17 MB). The original corpus is not needed for any of the following problems but is provided for those interested in a detailed analysis of their results. The original corpus has been preprocessed and divided into a training and a testing part, 20news.preproc.train and 20news.preproc.test. Finally, in 20news.preproc.vocabulary you find the vocabulary of all the texts. Without further explanation note that the order of the vocabulary is related to the importance of each word for classification and should not be altered for the experiments in the following. You will also find a solution framework in the directory.

See second page on reverse side!

(a) Assume the training data for each class (topic) is given by a set of texts $t_k = 1, \ldots, T_k$ which are represented by the corresponding word counts, i.e. the number $n_{t_k i}$ of occurences of each word w_i of the vocabulary in each text t. (* 6P)
Derive the *Maximum Likelihood* estimation of the corresponding multinomial distribution given the corresponding training data.

(b) Implement a *Maximum Likelihood* estimator for the multinomial distributions of all classes based on your result from ??, which takes the training data file 20news.preproc.train and some vocabulary vocab as input. The output of your estimator should be a single file with the parameters of the multinomials of all classes given. In addition the estimator should estimate the prior probabilties of all classes as relative frequencies and include these into the parameter output file. (* 5P)

(c) Implement an efficient *Bayesian* classifier using multinomial class conditional distributions and prior probabilities as estimated with the corresponding estimator implemented in ??. The input to the classifier will be the testing data file 20news.preproc.test as well as the corresponding parameter file provided by the estimator and the corresponding vocabulary file vocab. As output the classifier should produce an output file with one line for each classified text, where the first column gives the correct class given in the testing data file and the second column the recognized class obtained by your classifier. (* 5P)
Write also an evaluation tool which computes the error rate and the confusion matrix. The tool takes as input recognition results in the format given by the output of the above classifier.

(d) Now use your estimator from ?? to estimate the parameters using the training data 20news.preproc.train. Then use your classifier from ?? to classify the testing data file 20news.preproc.test using the vocabulary 20news.preproc.vocabulary reduced to the *first* V entries, with $V = 100, 200, \ldots, 1900, 2000, 3000, \ldots, 10000$. Plot the error rate as a function of vocabulary size V. (* 2P)

(e) Modify the training data 20news.preproc.train by adding, for each class, a fictive text containing each word of the vocabulary exactly once. Repeat the experiments from ?? using this modified training data and plot the error rate as function of vocabulary size together with your results from ??. Try to give an explanation for the change in results. (* 2P)