# Learning Vector Quantization, Multilayer Perceptron, Neurofuzzy Network and Logistic Regression in the Diagnosis of Prostate Cancer

Mehran Gomari*
Turku Centre for Computer Science
DataCity, Lemminkäisenkatu 14A
FIN-20520 Turku, Finland

Patrik Finne
Department of Clinical Chemistry
University Central Hospital of Helsinki
Haartmaninkatu 2
FIN-00270 Helsinki, Finland

Timo Järvi
Turku Centre for Computer Science
DataCity, Lemminkäisenkatu 14A
FIN-20520 Turku, Finland

Ulf-Håkan Stenman
Department of Clinical Chemistry
University Central Hospital of Helsinki
Haartmaninkatu 2
FIN-00270 Helsinki, Finland

Jonas Hugosson
Department of Urology
Sahlgrenska University Hospital
S-41345 Göteborg, Sweden

**Abstract** *Diagnosis of prostate cancer can be facilitated and the medical costs used for cancer detection can be lowered if a system which uses only laboratory tests can classify cancer and non-cancer cases with acceptable accuracy. In this study, logistic regression method and three kinds of artificial neural networks, namely multilayer perceptron, neurofuzzy network and learning vector quantization are investigated for predicting the biopsy results of a screening database of 536 men (97 cancer cases) with prostate specific antigen (PSA) values between 3 and 10 $\mu g/l$ from Gothenburg, Sweden. To obtain a realistic approximation of the models' performance in real use, cross-validation method is used. The best results were obtained by the learning vector quantization model: it predicted the biopsy results with an average of 77% overall accuracy, 72% sensitivity and 78% specificity. At sensitivity levels of 81% and 95% it provided 69% and 44% specificity, respectively. The classification ability of the learning vector quantization models were significantly better than that of the ratio of free PSA to total PSA ($p < 0.01$); it was also significantly better than that of the other models ($p < 0.01$). In conclusion, this study suggests that by utilizing the learning vector quantization models instead of the PSA or the ratio of free to total PSA, the clinical decision making for prostate cancer can be improved by nearly halving the false-positive results.*

*Keywords:* neural networks, logistic regression, prostate cancer, prostate specific antigen, sensitivity, specifity

## 1 Introduction

Prostate cancer is the most common non-skin cancer in males and it is the second most

*Corresponding author. Email: gomari@cs.utu.fi, Phone: +358 2 3338670, Fax: +358 2 3338600. Supported by the Technology Development Centre of Finland (TEKES) and EG&G Wallac.

common cause of cancer death in American men [1]. American Cancer Society estimated that only in the United States in 1997 about 334,500 new cases would be diagnosed and 41,800 men would die of this disease [1]. The most used screening method for detecting prostate cancer is measuring the concentration of prostate specific antigen (PSA) in blood. Using PSA alone for the diagnosis of prostate cancer does not provide a sufficient diagnostic accuracy. Measuring free PSA and using its ratio to total PSA (F/T-ratio) is suggested by some researchers to improve the diagnostic accuracy [2]. Although using F/T-ratio can improve the diagnostic accuracy, still more effective methods would be needed to decrease the number of false cases.

In a previous study [3], artificial neural networks (ANNs) have been applied for the diagnosis of prostate cancer, where an ANN provided considerably higher accuracy than PSA. However, the results were based on a single test set and comparison with the F/T-ratio or logistic regression was not performed. In addition only one type of ANNs was used in that study.

The objective of this study is to investigate the capability of a logistic regression model and three different types of ANNs for predicting the biopsy results based on some laboratory tests. The effectiveness of these methods is evaluated in a reliable manner and finally these results are compared with the capability of the F/T-ratio for the diagnosis of prostate cancer.

# 2 Material

The data used in this study were extracted from a prostate cancer screening database in Gothenburg (Sweden) comprising 608 men (137 cancer cases) with a serum PSA concentration over 3 $\mu$g/l. For all cases, the following information was available: age, total and free PSA, digital rectal examination (DRE) result, transrectal ultrasound examination (TRUS) result, symptoms of the disease, the history of the disease in the family, and the biopsy result. From this database, only those cases aged between 51–66 and with a PSA level between 3–10 $\mu$g/l, which diagnostically lie in the difficult area, were selected for the study sample. This study sample contained 536 cases of which 97 were cancer cases.

# 3 Methods

## 3.1 Logistic regression

Regression methods are commonly used to discover the relationship between explanatory or input variables and their dependent response or output. Logistic regression (LR) [4] is used for the binary output variables. LR differs from conventional linear regression methods in that it predicts a transformation of the output instead of the output itself. This is called logit transformation and is defined by:

$$logit(p) = log(\frac{p}{1-p})$$

where $p$ and $1-p$ are the probabilities of the two possible events. For instance when the response is the presence or absence of prostate cancer, $p$ can be interpreted as the probability of having cancer and $1-p$ as the probability of not having cancer.

## 3.2 Multilayer perceptron

Multilayer perceptron (MLP) [5, 6, 7, 8] is the most widely used feed-forward ANN for classification problems. MLPs consist of an input layer, an output layer and one or more hidden layers, where each layer consists of a number of *units* or *neurons*. Units are connected by *connections* or *links* and an adaptive weight is associated with each connection. Given a pair of input-output vectors, MLPs can learn to represent the functional relationship between inputs and outputs. Learning takes place by adjusting the weights so that an error function is minimized. This error function is usually the mean square error (MSE) which is defined as follows:

$$E = \frac{1}{P} \sum_{p=1}^{P} (\bar{t}_p - \bar{o}_p)^2$$

where $P$ is the number of input-output patterns, $\bar{t}_p$ is the $p$th target output and $\bar{o}_p$ is the $p$th computed output by the network.

### 3.3 Neurofuzzy network

Neurofuzzy networks (NFNs) [9] incorporate the properties of ANNs and fuzzy logic [10, 11]. In contrast to many feed-forward ANN architectures, such as MLPs, NFNs store information locally. Typical NFN consists of an input, an output and a hidden layer. Each layer contains one or more units and a *basis function* is associated with each hidden unit. These basis functions which are defined on the normalized input space, force the network to represent the information locally. The basis functions in NFNs are used to represent fuzzy membership functions [11]. The fuzzy membership functions implement fuzzy linguistic terms such as *small, very small* or *almost zero* which in turn are used to generate fuzzy rules such as:

IF PSA is *small* AND F/T-ratio is *very high*
THEN it is *more likely* that the person
does not have prostate cancer.

Such fuzzy production rules can be used for modeling or control. NFNs are *transparent* in the sense that the internal representation of the model can be interpreted using the fuzzy linguistic production rules. This is a major advantage over the conventional ANNs whose internal representation is usually *opaque*.

### 3.4 Learning vector quantization

Vector quantization (VQ) [12] was originally designed for data compression whereby instead of storing all data points (vectors), one may store *m templates* or *code vectors* and later use an encoded version of the templates to represent the data vectors.

Learning vector quantization (LVQ) [13] is a supervised learning method for classification problems. The class information is used to construct the code vectors in a reward-punishment manner. First a subset of the code vectors is assigned to each class and then a learning algorithm (LVQ1, LVQ2, LVQ3 or OLVQ1 [13]) is used to fine-tune the code vectors by minimizing the rate of misclassification errors. For each new sample the nearest-neighbor rule according to the Euclidean distance is used to define the winner code vector. During the learning process the winner is either rewarded (if the sample and winner belong to the same class) or punished (if the sample and winner belong to different classes) according to some adaptation rules.

Since for classification problems it is the class borders and not the inside of the class regions that are of most importance, it suffices for the LVQ algorithm to place the code vectors into the input space in such a way that the nearest-neighbor rule minimizes the average misclassification error.

## 4 Techniques

### 4.1 Selecting the input variables

For those methods that had an automatic variable selection procedure, the input variables were selected from all available input variables using that procedure. Thus using the stepwise variable selection procedure, the LR method selected F/T-ratio, DRE and TRUS, while the NFN method selected total and free PSA, DRE and TRUS using its automatic network construction tool. However, for the MLP and LVQ methods such a procedure was not available, moreover they are supposed to handle redundant variables. Therefore for the MLP and LVQ experiments we kept all available input variables except age which was set aside due to some preliminary tests which showed that the overall performance was higher when age was discarded.

## 4.2 Normalization

An important issue in training a system using a learning algorithm is rescaling or normalizing of the data. For many learning algorithms raw data cannot be used without rescaling and in many cases rescaling improves the performance. However, rescaling may also cause loss of information. A major problem in rescaling is that if the training inputs are measured with different scales, there would not be a simple way of rescaling. Another problem with rescaling is the dependency between the variables. Simple linear rescaling methods treat the variables as independent although they frequently are dependent. One way to overcome this problem is to use a more sophisticated linear rescaling, called whitening [8], which allows also for correlation between the variables.

In this study sample, except total and free PSA all the other variables were nominal, therefore rescaling did not apply for them. For the total and free PSA, which were highly correlated, we used whitening procedure for the three ANN methods. However, for the LR experiments, because of a positive skewness of these two variables, a logarithmic transformation was employed as one approach before selecting the input variables.

## 4.3 Early and optimal stopping

A substantial issue in building a decision support system is its reliability in real use, i.e. the system must perform on novel cases as well as on the training cases. ANNs suffer from the problem of *overfitting* (*overtraining*), i.e. if the training continues beyond an optimal stopping point for generalization, ANNs tend to learn the specific features of the training set while losing their generalizing capability. However, it is difficult to define this optimal stopping point.

One approach for solving the problem leaves out a portion of training data to be used as a validation set. The training will be stopped at the point where the validation set's error starts to increase. This approach is called early stopping [8]. Although this method can prevent the network from overtraining, it can on the other hand cause inadequate training, because there may be many local minima in the error space and that the first found minimum may not be the optimal one. To solve this problem we can continue the training process until we find the optimal minimum according to the validation set. Practically, this is achieved by continuing the training for a long enough time and always keeping record of the network's state at the minimum point reached. We call this approach as optimal stopping. In this study we used the optimal stopping method for the MLP and LVQ examinations.

## 4.4 Cross-validation

A simple way to approximate the model's performance in real use is to separate a part of the data as the test set. The test set is unseen during the training process and will be used to approximate the model's performance after the training phase. However, this is not an efficient way since this method suffers from sampling variation, moreover we may not want to set aside a part of the valuable data only for comparison. To get rid of this problem one can divide the available data at random into K equal and distinct parts. Further, one part is used as the test set while other parts together are treated as the training set. This process is repeated until each part has been used once as the test set. The final result is the average over all test sets. This technique is called K-fold cross-validation [8].

## 4.5 Practical considerations

### 4.5.1 Logistic regression

The 10-fold cross-validation method was employed to evaluate the LR results. Two different approaches were examined for prepro-

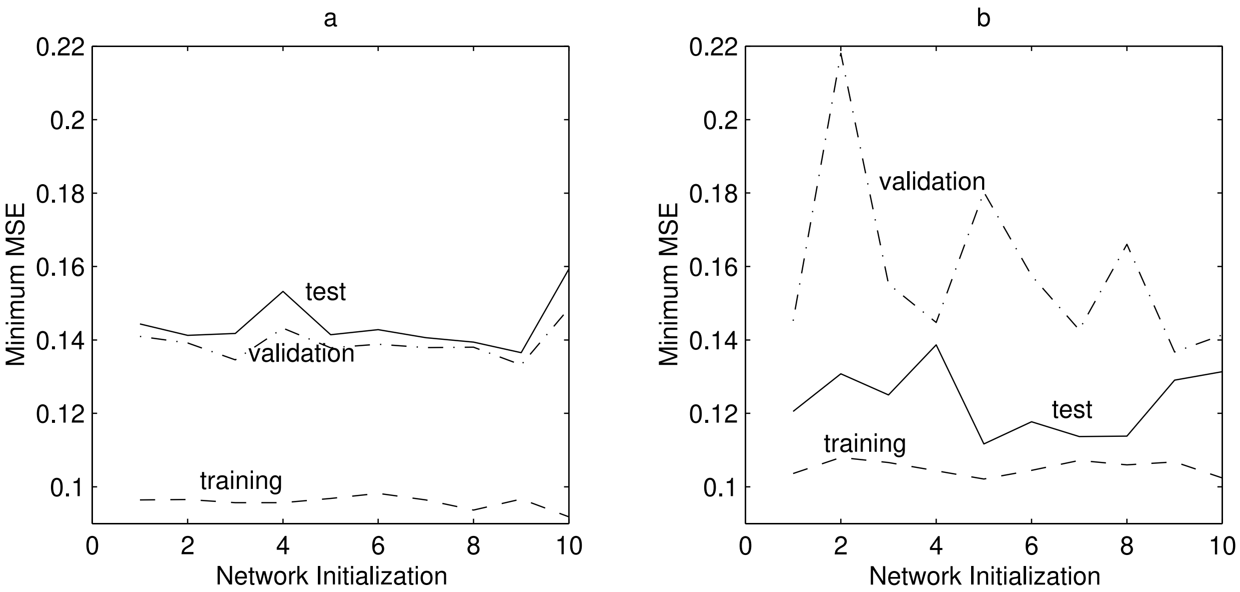


Figure 1: Minimum MSE after 300 training epochs for training, validation and test sets while 10 different initial weights were used. a) The validation and test sets resemble each other and they reach their minimum with the same weight initialization. Therefore choosing the minimum MSE according to the validation set ensues the minimum MSE for the test set, too. b) The test set seems to resemble the training set rather than the validation set. Thus choosing the minimum MSE according to the validation set does not ensue the minimum MSE for the test set.

cessing of the total and free PSA. In the first one a logarithmic transformation of these two variables was performed, whereas in the second one they were replaced by the F/T-ratio. The other input variables were included in both examinations. Forward and backward stepwise logistic regressions were executed for both sets of input variables and the two stepwise methods gave the same results. Age, disease symptoms and heredity were not significant ($p > 0.05$) and were therefor excluded. When at least 75% sensitivity was required, the model with the F/T-ratio produced slightly higher specificity in the training sets; thus we selected it as the final model. The model's equation was:

$$logit(p) = a + b \times \frac{F}{T} + c \times DRE + d \times TRUS$$

where the coefficients $a, b, c$ and $d$ varied depending on the training set used. These experiments were performed with SPSS 6.1.1 for Macintosh.

#### 4.5.2 Multilayer perceptron

Theoretically, an MLP with one hidden layer is capable of approximating any classifier-type mapping [7]. However this approximation depends on many adjustable parameters which usually have to be found by experimentation. In this study, the number of hidden layers and units for the final MLP model was chosen based on some preliminary experiments. These experiments showed that an MLP with five hidden units in one hidden layer could attain almost the same efficiency as the MLPs with more hidden units. Since we prefer simplicity of the model to complexity, we decided to choose the MLP with five hidden units in one hidden layer as the final model.

To organize a fair evaluation of the model's performance on unseen data and at the same time to avoid overfitting, the 10-fold cross-validation method was combined with the optimal stopping method. Practically it means that in addition to the test set, another part was chosen randomly as the validation set and the remaining eight parts together formed the training set. In this way the training sets became smaller than for the LR model, however the same test sets were used. For each cross-validation trial, the model was first trained using the training set, trying to minimize the mean square error function according to the validation set, after which the model's performance was evaluated using the test set. This technique works well providing that the validation and test set resemble each other and do not contain inconsistent data. Figure 1 illustrates two typical situations encountered while using this technique.

As mentioned in section 4.3, in practice it is difficult to define when the minimum error of the validation set is achieved. Therefore we always kept a record of the network's state at the last observed minimum error while the training continued for a longer time.

The learning algorithm used for the MLP experiments was a faster version of the back-propagation algorithm, called resilient back-propagation (Rprop) [14]. The network was initialized with random weights between (-1, 1). The training was performed ten times with different initialization of the weights to find the optimal initial weights. Furthermore, two different activation functions, symmetric sigmoid (-0.5, 0.5) and hyperbolic tangent were examined. The MLP experiments were performed by the Stuttgart Neural Network Simulator (SNNS) version 4.1[1].

#### 4.5.3 Neurofuzzy network

The learning algorithm used for the NFN experiments was Adaptive Spline Modeling of Observation Data (ASMOD) [9]. Since the ASMOD algorithm is an automatic network construction algorithm, there was no need for a validation set to terminate the training process. Therefore, As for the LR model, the 10-fold cross-validation was used for evaluating the NFN results. In other words, the same training sets as for the LR model were used for the NFN experiments although with different normalization.

An average of 11 rules in 2–3 subnetworks was used during the training process. For the NFN experiments, a simulator called Neuframe (NCS Manufacturing Intelligence, Southampton, UK) version 3.0 was used.

#### 4.5.4 Learning vector quantization

To find the optimal number of code vectors, a number of LVQ models with different number of code vectors ranging from 10 to 100 was examined. Based on these examinations the number of code vectors for the final model was chosen to be 50. Four different LVQ algorithms (LVQ1, LVQ2, LVQ3 and OLVQ1) from the Learning Vector Quantization Program Package (LVQ-PAK version 3.1, Helsinki University of Technology [15]) were employed. As for the MLP tests, the 10-fold cross-validation with optimal stopping

[1] More information can be found in the web site: http://www.informatik.uni-stuttgart.de/ipvr/bv /projekte/snns/snns.html

Table 1: Results for the test sets of the 10-fold cross-validations when at least 75% sensitivity was required in the training or validation sets. Overall classification accuracy, sensitivity, specificity, positive predictive value, negative predictive value and information gain are shown for the NFN, LR, MLP and LVQ models, respectively.

| Test | Acc. (%) | | | | Sens. (%) | | | | Spec. (%) | | | | PPV (%) | | | | NPV (%) | | | | IG (bits) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| set | NF | LR | ML | LV | NF | LR | ML | LV | NF | LR | ML | LV | NF | LR | ML | LV | NF | LR | ML | LV | NF | LR | ML | LV |
| 1 | 19 | 19 | 81 | 61 | 100 | 100 | 80 | 80 | 0 | 0 | 82 | 57 | 19 | 19 | 50 | 30 | 0 | 0 | 95 | 93 | 0 | 0 | 0.2 | 0.1 |
| 2 | 19 | 19 | 61 | 63 | 100 | 100 | 80 | 90 | 0 | 0 | 57 | 57 | 19 | 19 | 30 | 32 | 0 | 0 | 93 | 96 | 0 | 0 | 0.1 | 0.1 |
| 3 | 30 | 31 | 67 | 81 | 100 | 100 | 90 | 80 | 14 | 16 | 61 | 82 | 21 | 21 | 35 | 50 | 100 | 100 | 96 | 95 | 0 | 0 | 0.1 | 0.2 |
| 4 | 35 | 85 | 63 | 69 | 100 | 100 | 100 | 100 | 20 | 82 | 55 | 61 | 22 | 56 | 33 | 37 | 100 | 100 | 100 | 100 | 0 | 0.4 | 0.2 | 0.2 |
| 5 | 83 | 83 | 80 | 87 | 100 | 100 | 80 | 60 | 80 | 80 | 80 | 93 | 53 | 53 | 47 | 67 | 100 | 100 | 95 | 91 | 0.3 | 0.3 | 0.2 | 0.2 |
| 6 | 74 | 70 | 46 | 91 | 100 | 80 | 80 | 90 | 68 | 68 | 39 | 91 | 42 | 36 | 23 | 69 | 100 | 94 | 89 | 98 | 0.3 | 0.1 | 0 | 0.3 |
| 7 | 76 | 78 | 69 | 91 | 50 | 50 | 60 | 80 | 82 | 84 | 70 | 93 | 38 | 42 | 32 | 73 | 88 | 88 | 89 | 95 | 0 | 0.1 | 0 | 0.3 |
| 8 | 79 | 89 | 75 | 91 | 44 | 56 | 78 | 100 | 86 | 95 | 75 | 89 | 40 | 71 | 39 | 64 | 88 | 91 | 94 | 100 | 0 | 0.2 | 0.1 | 0.4 |
| 9 | 81 | 83 | 36 | 64 | 33 | 22 | 89 | 22 | 91 | 95 | 25 | 73 | 43 | 50 | 20 | 14 | 87 | 86 | 92 | 82 | 0 | 0 | 0 | 0 |
| 10 | 85 | 85 | 83 | 75 | 11 | 11 | 0 | 22 | 100 | 100 | 100 | 86 | 100 | 100 | 0 | 25 | 84 | 84 | 83 | 84 | 0 | 0 | 0 | 0 |
| *Ave.* | 58 | 64 | 66 | 77 | 74 | 72 | 74 | 72 | 54 | 62 | 64 | 78 | 40 | 47 | 31 | 46 | 75 | 74 | 93 | 93 | 0.1 | 0.1 | 0.1 | 0.2 |

method was used for the LVQ experiments.

In addition to the LVQ examinations with at least 75% sensitivity (percentage of correctly classified cancer cases) requirement in the validation sets; two other examinations were performed when a sensitivity of at least 80% was required. One with the same data sets as for the previous examinations and another with duplicated number of cancer cases in the training sets. The later was done to attain a higher sensitivity level.

### 4.6 Evaluation of the results

The results of the final models of different methods for the same test sets were evaluated using the following efficiency measures: overall classification accuracy, sensitivity, specificity, positive predictive value (PPV) and negative predictive value (NPV). In addition, an information theoretic [16] approach was used to measure the amount of information gain obtained by each model. The classification ability of the models was compared using McNemar's significance test [17].

## 5 Results

Table 1 displays the results for the NFN, LR, MLP and LVQ models when at least 75% sensitivity was required in the training (for the NFN and LR models) or validation (for the MLP and LVQ models) sets. On average, the LR and MLP models attained higher overall accuracy and specificity than the NFN model; and the MLP model produced slightly higher results than the LR model except for the PPV. There was no significant difference between the classification ability of the LR and MLP models, but their classification ability was significantly better than that of the NFN model (McNemar's test, $p < 0.01$).

The best results were obtained by the LVQ model. Although the LVQ model obtained slightly lower average sensitivity than the MLP and NFN models and slightly lower average PPV than the LR model, its average overall accuracy, specificity and the information gain were considerably higher. McNemar's tests for classification ability indicated that the LVQ model provided significantly higher accuracy than the NFN, LR and MLP models ($p < 0.01$).

Table 2: The 10-fold cross-validation results for the NFN, LR, MLP and LVQ models when at least 75% sensitivity was required in the training or validation sets. Overall classification accuracy, sensitivity, specificity, positive and negative predictive values and information gain are shown. Also the corresponding values for the F/T-ratio with 75% sensitivity are presented. The LVQ model attained 20 percent units higher specificity than the F/T-ratio whereas its sensitivity was only 3 percent units lower.

| Classifier | Acc. (%) | Sens. (%) | Spec. (%) | PPV (%) | NPV (%) | IG (bits) |
|---|---|---|---|---|---|---|
| NFN | 58 | 74 | 54 | 40 | 75 | 0.09 |
| F/T-ratio | 61 | 75 | 58 | 29 | 91 | 0.05 |
| LR | 64 | 72 | 62 | 47 | 74 | 0.12 |
| MLP | 66 | 74 | 64 | 31 | 93 | 0.09 |
| LVQ | 77 | 72 | 78 | 46 | 93 | 0.18 |

Table 3: Results for the test sets of the 10-fold cross-validations when at least 80% sensitivity was required in the validation sets. Overall classification accuracy, sensitivity, specificity, positive predictive value, negative predictive value and information gain are shown for the LVQ model using the original training sets and using the training sets with duplicated number of cancer cases, respectively.

| Test | Acc. (%) | | Sens. (%) | | Spec. (%) | | PPV (%) | | NPV (%) | | IG (bits) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| set | Orig. | Dupl. | Orig. | Dupl. | Orig. | Dupl. | Orig. | Dupl. | Orig. | Dupl. | Orig. | Dupl. |
| 1 | 61 | 52 | 80 | 90 | 57 | 43 | 30 | 26 | 93 | 95 | 0.06 | 0.06 |
| 2 | 56 | 54 | 80 | 80 | 50 | 48 | 27 | 26 | 92 | 91 | 0.04 | 0.04 |
| 3 | 81 | 69 | 80 | 100 | 82 | 61 | 50 | 37 | 95 | 100 | 0.19 | 0.22 |
| 4 | 69 | 67 | 100 | 90 | 61 | 61 | 37 | 35 | 100 | 96 | 0.22 | 0.13 |
| 5 | 83 | 69 | 90 | 100 | 82 | 61 | 53 | 37 | 97 | 100 | 0.25 | 0.22 |
| 6 | 83 | 48 | 90 | 100 | 82 | 36 | 53 | 26 | 97 | 100 | 0.25 | 0.11 |
| 7 | 91 | 87 | 80 | 100 | 93 | 84 | 73 | 59 | 95 | 100 | 0.30 | 0.38 |
| 8 | 91 | 55 | 100 | 100 | 89 | 45 | 64 | 27 | 100 | 100 | 0.41 | 0.13 |
| 9 | 25 | 15 | 89 | 89 | 11 | 0 | 17 | 15 | 83 | 0 | 0.0 | 0.05 |
| 10 | 75 | 17 | 22 | 100 | 86 | 0 | 25 | 17 | 84 | 0 | 0.01 | 0.0 |
| *Ave.* | 71 | 53 | 81 | 95 | 69 | 44 | 42 | 31 | 94 | 78 | 0.17 | 0.13 |

Table 4: The 10-fold cross-validation results for the LVQ model when at least 80% sensitivity was required and the corresponding values for the F/T-ratio with the same sensitivity levels. The LVQ results for the original training sets and for the training sets with duplicated number of cancer cases are shown respectively. Overall classification accuracy, sensitivity, specificity, positive and negative predictive values and information gain are shown.

| Classifier | Acc. (%) | | Sens. (%) | | Spec. (%) | | PPV (%) | | NPV (%) | | IG (bits) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Orig. | Dupl. | Orig. | Dupl. | Orig. | Dupl. | Orig. | Dupl. | Orig. | Dupl. | Orig. | Dupl. |
| LVQ | 71 | 53 | 81 | 95 | 69 | 44 | 42 | 31 | 94 | 78 | 0.17 | 0.13 |
| F/T-ratio | 57 | 32 | 81 | 95 | 52 | 18 | 27 | 20 | 93 | 94 | 0.05 | 0.02 |

The cross-validation results for the four models along with the appropriate results for the F/T-ratio are shown in table 2. Comparing these models with the performance of the F/T-ratio for the diagnosis of prostate cancer, it appears that the LR, MLP and LVQ models provided higher scores whereas the F/T-ratio outperformed the NFN model. On average, The LVQ model obtained 16 percent units (p.u.) higher accuracy, 20 p.u. higher specificity, 17 p.u. higher positive predictive value, 2 p.u. higher negative predictive value and 0.13 bits higher information gain than the F/T-ratio, while its sensitivity was only 3 p.u. lower. There was no significant difference between the classification ability of the F/T-ratio and the LR or MLP model, but The classification ability of the LVQ model was significantly better than that of the F/T-ratio (McNemar's test, $p < 0.01$).

When at least 80% sensitivity was required in the validation sets, a higher sensitivity was also obtained for the test sets . Moreover, duplication of the cancer cases in the training sets resulted in a 14 p.u. increase in the average sensitivity level of the test sets. The results for these experiments are shown in table 3. These results show that by increasing the sensitivity level, the specificity and consequently the overall accuracy (because of higher frequency of non-cancer cases) decrease.

The cross validation results for these experiments along with the corresponding values for the F/T-ratio with the same sensitivity levels are shown in table 4. The LVQ model with the original training sets attained 17 p.u. higher specificity whereas the one with duplicated number of cancer cases attained 26 p.u. higher specificity. The lower negative predictive value for the LVQ model with duplicated number of cancer cases is due to the fact that no non-cancer case was recognized in the last two test sets. The classification ability of the LVQ models with the average sensitivity level of 81% and 95% were significantly better than that of the F/T-ratio with equal sensitivity levels (McNemar's test, $p < 0.01$).

## 6 Conclusion

This study indicated that there was no significant difference between the LR and MLP models. Furthermore it suggests that the LVQ models are more suitable for classification problems, since the class borders can be optimized to minimize the misclassification error even if the class distribution of the input patterns overlap.

By utilizing the LVQ model instead of the F/T-ratio, it is possible to improve the clinical decision for the diagnosis of prostate cancer by nearly halving the number of false-positive cases (22% false-positive rate versus 42%). Practically this could help the physicians by avoiding unnecessary further investigation of the disease for the false-positive cases.

## References


[1] American Cancer Society. American Cancer Society Surveillance Research. 1997.

[2] U.H. Stenman, M. Hakama, P. Knekt, A. Aromaa, L. Teppo, and J. Leinonen. Serum concentration of prostate specific antigen and its complex with $\alpha_1$-antichymotrypsin before diagnosis of prostate cancer. *The Lancet*, 344:1594–1598, 1994.

[3] P.B. Snow, D.S. Smith, and W.J. Catalona. Artificial Neural Networks in the Diagnosis and Prognosis of Prostate Cancer: A Pilot Study. *Journal of Urology*, 152:1923–1926, 1994.

[4] D.W. Hosmer and S. Lemeshow. *Applied Logistic Regression.* John Wiley & Sons, New York, 1989.

[5] D.E. Rumelhart, J.L. McCelland, and PDP research group. *Parallel Distributed Processing, Exploration in the*

*Microstructure of Cognition, Vol.1: Foundations.* MIT Press, Cambridge, Massachusetts, 1986.

[6] J.A. Freeman and D.M. Skapura. *Neural Networks: Algorithms, Applications, and Programming Techniques.* Addison-Wesley Publishing Company, U.S.A., 1991.

[7] M. H. Hassoun. *Fundamentals of Artificial Neural Networks.* The MIT Press, Cambridge, Massachusetts, London, 1995.

[8] C. M. Bishop. *Neural Network for Pattern Recognition.* Clarendon Press, Oxford, 1995.

[9] M. Brown and C. Harris. *Neurofuzzy Adaptive Modelling and Control.* Prentice Hall, New York, 1994.

[10] L.A. Zadeh. Fuzzy Sets. *Information and Control*, 8:338–353, 1965.

[11] L.A. Zadeh. Outline of a new approach to the analysis of the complex systems and decision processes. *IEEE Trans. on Systems, Man and Cybernetics*, 1:28–44, 1973.

[12] A. Gersho and R.M. Gray. *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, Boston, Dordrecht, London, 1992.

[13] T. Kohonen. *Self-Organizing Maps.* Springer-Verlag, Berlin, Heidelberg, 1995.

[14] M. Riedmiller and H. Braun. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In *Proc. of IEEE International Conference on Neural Networks*, pages 588–591. IEEE, 1993.

[15] T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, and K. Torkkola. Lvq_pak: The learning vector quantization program package. Technical report, Helsinki University of Technology, Otaniemi, Finland, 1996.

[16] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication.* University of Illinois Press, Urbana, 1949.

[17] L. Sachs. *Applied Statistics, A Handbook of Techniques.* Springer-Verlag, New York, 1984.