Comments and questions can be sent to: michael@cs.uu.nl

ELSEVIER

Pattern Recognition Letters 20 (1999) 1439–1448

Pattern Recognition Letters

www.elsevier.nl/locate/patrec

Sequential selection of discrete features for neural networks – A Bayesian approach to building a cascade

M. Egmont-Petersen [a,*], W.R.M. Dassen [b], J.H.C. Reiber [a]

[a] *Department of Radiology, Division of Image Processing (LKEB), Leiden University Medical Center, P.O. Box 9600, 2300 RC Leiden, The Netherlands*

[b] *Department of Cardiology, Academic Hospital Maastricht, Maastricht, The Netherlands*

Abstract

A feature selection procedure is used to successively remove features one-by-one from a statistical classifier by an iterative backward search. Each classifier uses a smaller subset of features than the classifier in the previous iteration. The classifiers are subsequently combined into a cascade. Each classifier in the cascade should classify cases to which a reliable class label can be assigned. Other cases should be propagated to the next classifier which uses also the value of a new feature. Experiments demonstrate the feasibility of building cascades of classifiers (neural networks for prediction of atrial fibrillation (FA)) using a backward search scheme for feature selection. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Cascade; Feature selection; Feature assessment; Sequential classification; Rejection; Backward search; Pruning; Atrial fibrillation

1. Introduction

In the fields of pattern recognition and image processing, statistical classifiers are often trained to perform classification tasks such as texture segregation (Egmont-Petersen and Pelikan, 1999), image segmentation (Tian et al., 1999) or object recognition. In general, a large number of features is available or can be defined, although often only a few features determine the most likely class membership of most cases or pixels. For such applications, it is desired to prune redundant features and thus possibly decrease the acquisition costs (e.g., computational costs) associated with measuring the features.

We will present an approach that uses a feature selection procedure to construct a so-called *cascade* of statistical classifiers. Our goal is to reduce the average feature acquisition costs per case by classifying subsets of cases using as few features as possible in a multistage classification scheme (Pudil et al., 1992). For that purpose, we first build a classifier (a feed-forward neural network) using all features and prune those that are completely redundant. The remaining n features are subsequently pruned one-by-one according to a backward search scheme. Each step results in a classifier with one feature less than its predecessor. Pruning can be continued until all n classifiers, using $1, 2, \ldots, n$ input values, have been built. The resulting statistical classifiers are combined into a

* Corresponding author. Tel.: +31-71-526-2285; fax: +31-71-524-8256.

E-mail address: michael@lkeb.azl.nl (M. Egmont-Petersen)

0167-8655/99/$ - see front matter
PII: S0167-8655(99)00112-9

cascade which constitutes a statistical classifier in itself. The first classifier in the cascade obtains as input vector the minimally required set of feature values (e.g., 1 feature value) and should, if possible, only assign reliable class labels to the cases. Other cases should be left unclassified and propagated to the next classifier in the cascade. The subsequent classifier requires the value of one more feature to decide whether reliable class labels can be assigned to the propagated cases.

2. Background

Classification entails assigning a class label to a case characterized by an n-dimensional feature vector $\boldsymbol{x}$. Let $p(\boldsymbol{x}|\omega_j)$ denote the n-dimensional class-conditional probability density function (PDF) for class j when the n features are continuous and let $p(\boldsymbol{x}|\omega_j)$ denote the n-dimensional class-conditional probability function (PF) when the features are discrete. [1] In general, classifiers partition the feature space into disjoint regions R_j, $j = 1, \ldots, c$, with c denoting the number of classes. For a minimal error-rate classifier, cases $\boldsymbol{x}$ that occur in the region

$$R_j^n = \{\boldsymbol{x} \in \mathrm{rng}(\boldsymbol{x}) | P(\omega_j)P(\boldsymbol{x}|\omega_j) > P(\omega_i)P(\boldsymbol{x}|\omega_i)\ \forall i \neq j\} \quad (1)$$

have the highest posterior probability of belonging to class j and are classified as such. The function $\mathrm{rng}(\boldsymbol{x})$ denotes the range of $\boldsymbol{x}$. Define the *correctness* of a classifier based on n features as

$$\rho^n = \sum_{j=1}^{c} P(\omega_j)P(\boldsymbol{x} \in R_j^n|\omega_j) \quad (2)$$

and the *marginal contribution* of feature k as

$$\Delta\rho^{\neq k} = \sum_{j=1}^{c} P(\omega_j)\Big(P(\boldsymbol{x} \in R_j^n|\omega_j) - P(\boldsymbol{x}^{\neq k} \in R_j^{n-1}|\omega_j)\Big), \quad (3)$$

[1] Henceforth, only classification problems with discrete features are considered. For a treatment of continuous features, see (Egmont-Petersen et al., 1998).

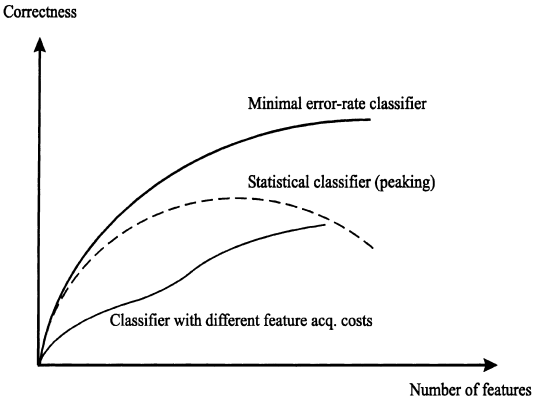

Fig. 1. The three curves indicate the correctness (fraction of correctly classified cases) of the minimal error-rate classifier, of a statistical classifier that exhibits peaking and of a classifier where features are pruned while the (different) acquisition costs of the features are taken into account.

with $\boldsymbol{x}^{\neq k}$ denoting an $n-1$ dimensional feature vector that is equal to $\boldsymbol{x}$ except for feature k that has been removed, and R_j^{n-1} the Bayes optimal region similar to R_j^n but defined for the $n-1$ dimensions excluding dimension k.

When features are pruned one-by-one from a minimal error-rate classifier according to their marginal contribution, e.g., using the LMS-pruning scheme (Egmont-Petersen et al., 1998), the correctness is a monotonous, concave function of the number of features used (see 'Minimal error-rate classifier' in Fig. 1). Removing features always leads to a decrease in correctness, $\Delta\rho^{\neq k} \geqslant 0$. The correctness of statistical classifiers fitted on a training set, however, might increase, $\Delta\rho^{\neq k} < 0$, because of overgeneralization. Experiments have shown that contrary to the k-nearest neighbor and Parzen window classifiers, neural networks are unlikely to exhibit peaking (Hamamoto et al., 1996), a quality which makes backward search a suited feature selection scheme for neural networks. Another advantage of a backward search scheme is that it takes all mutual dependencies between features into account.

3. A cascade of classifiers

A set of classifiers obtained from a backward search feature selection procedure can be combined

into a cascade by first pruning completely superfluous features. Subsequently, the features that have influence on the performance of the classifier are pruned one-by-one. Which features should be pruned depends on the costs of different misclassifications and the acquisition costs of the features.

3.1. Pruning superfluous features

The first step in any feature selection procedure entails removing redundant features. Whether a feature is redundant, depends on which other features are provided as input to the classifier. We have earlier presented a feature measure called *potential influence* (Egmont-Petersen et al., 1998) which is the probability that the value of feature k can possibly determine which classification is assigned to a case

$$\phi_k \equiv \rho^n - \sum_{j=1}^{c} P(\omega_j) \times \sum_{\boldsymbol{x}^{\neq k} \in R_j^{n\setminus k}} P(\boldsymbol{x}^{\neq k}|\omega_j) g\big(S_j(\boldsymbol{x}^{\neq k}) = \mathrm{rng}(x_k)\big), \quad (4)$$

with

$$S_j(\boldsymbol{x}^{\neq k}) = \left\{ x_k \in \mathrm{rng}(x_k) \left| \frac{P(x_k|\boldsymbol{x}^{\neq k}, \omega_j)}{P(x_k|\boldsymbol{x}^{\neq k}, \omega_i)} > \frac{P(\omega_i)P(\boldsymbol{x}^{\neq k}|\omega_i)}{P(\omega_j)P(\boldsymbol{x}^{\neq k}|\omega_j)} \ \forall i \neq j \right. \right\}, \quad (5)$$

the set of x_k (for given $\boldsymbol{x}^{\neq k}$) for which $\boldsymbol{x}$ falls into R_j^n. $R_j^{n\setminus k}$ is the set of $\boldsymbol{x}^{\neq k}$ for which $S_j(\boldsymbol{x}^{\neq k})$ is not empty, see (Egmont-Petersen, 1996). The function $g(e)$ is 1 when the expression e is True, otherwise $g(e)$ is 0. As proven by Egmont-Petersen et al. (1998), the potential influence of a feature is a lower bound for the correctness that can be obtained from a classifier after this feature has been removed. The potential influence of a redundant feature is zero.

3.2. Pruning relevant features

After all redundant features have been removed, the n (remaining) features that actually influence the classification of cases are subsequently pruned one-by-one according to a backward search scheme. The decrease in correctness $\Delta\rho^{\neq k}$ that results from pruning a feature depends on the *modus* of classification task: Quinlan (1993) distinguishes between *sequential* and *parallel* classification tasks. In parallel classification tasks, all features are relevant for the classification of each case. In sequential classification tasks, only a few features are relevant for the classification of each case. Whether the remaining features are relevant for the classification of a case depends on the values of one or more of the other features. Sequential and parallel classification tasks form two ends of a continuum. For a classification task that is primarily parallel, pruning only a few features will lead to a large decrease in performance, i.e., the marginal contribution $\Delta\rho^{\neq k}$, $k = 1, \ldots, n$, of each feature is relatively high. For a sequential classification task, on the other hand, some features are likely to have small marginal contributions.

The influence of a feature on classifier performance is not always an optimal assessment criterion for feature selection. Assuming that a class label should eventually always be assigned to each case, three situations can be discerned:

1. All acquisition costs are equal and all misclassifications imply the same loss. A misclassification should always be avoided when possible.
2. The acquisition costs differ but all misclassifications imply the same loss.
3. The acquisition costs differ and each type of misclassification is associated with a specific loss.

In the first situation, the best assessment criterion is the marginal contribution of a feature, $\Delta\rho^{\neq k}$. When the acquisition costs differ, one has to make a trade-off between the marginal contribution of a feature, its acquisition costs and the costs of misclassifying cases. In the third situation, the assessment criterion should be based on the marginal risk of the classifier computed using a cost matrix $\boldsymbol{K}$, see (Duda and Hart, 1973). The first and second situations will be analyzed here; the third situation is considered by Pudil et al. (1992).

3.3. Pruning relevant features – equal acquisition costs

In this situation, one wants to prune features one-by-one according to their marginal contribution. This entails building a classifier from all n features, computing the marginal contribution of each feature, $\Delta\rho^{\neq k}$, and pruning the feature which leads to the smallest decrease in performance when removed from the classifier. This procedure can be repeated until only one feature remains.

3.4. Pruning relevant features – different acquisition costs

When the acquisition costs of features differ, it is necessary to specify the trade-off between a misclassified case and the costs of measuring each feature. Let $c(x_k)$ denote the costs associated with measuring feature k for a case. The trade-off between acquisition costs and the marginal contribution for two successive classifiers, one with and the other without feature k, we call the *marginal utility* of a feature

$$\gamma_k \equiv \Delta\rho^{\neq k} c(\overline{\omega}|\omega) - P(\varnothing)c(x_k) \tag{6}$$

with $c(\overline{\omega}|\omega)$ denoting the costs of a misclassified case and $P(\varnothing)$ the probability that a case is left unclassified by the first of the two classifiers. When γ_k is used as the pruning criterion in a backward search, features with a negative marginal utility should be abandoned, as the resulting increase in correctness yielded by adding the feature to the classifier is not compensated by the excess measurement costs. Although the (minimal error-rate) classifier that uses all n features will have the maximal correctness, the assumption that the correctness is a concave function of the number of features used does not hold when the marginal utility γ_k is used as the pruning criterion (see Fig. 1).

3.5. Building a cascade of classifiers

When the features have been pruned one-by-one by a backward-search scheme, the n classifiers can be combined into a so-called *cascade*. Its classifiers in concert are able to classify cases using increasingly larger subsets of features. In the sequel, we define a cascade of classifiers $\boldsymbol{C}$. Let $\boldsymbol{v}$ denote an n-dimensional vector indicating which features are needed by a particular classifier in the cascade, $v_k = 1$ when feature k is required as input and $v_k = 0$ when feature k is not. Define the indicator matrix $\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n)$, with $\sum_{i=1,\ldots,n} v_{i,l} = l$. $\boldsymbol{V}$ specifies the subsets, 1 to n, of features used by a cascade of statistical classifiers. Define a Bayesian cascade

$$\boldsymbol{C} = \{\boldsymbol{V}, \boldsymbol{B}, \boldsymbol{Z}\}, \tag{7}$$

with $\boldsymbol{B} = (b_1, b_2, \ldots, b_n)$ indicating the ordered set of (cascaded) classifiers and $\boldsymbol{Z}$ a set of discriminant functions used to assign a class label from the vector of posterior probabilities $\boldsymbol{o}_l$, $o_{j,l} = P^l(\omega_j|\boldsymbol{x}^{\neq v_l})$. The vector $\boldsymbol{x}^{\neq v_l}$ contains the features used as input to classifier l. $\boldsymbol{Z}$ could, e.g., be defined as a set of thresholds, $\boldsymbol{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{n-1}\}$, with each vector $\boldsymbol{z}_l$ pertaining to a particular classifier in the cascade and element $z_{j,l}$ the threshold used by the classification rule

$$L(\boldsymbol{o}_l) = \begin{cases} j: & o_{j,l} - o_{i,l} > z_{j,l}, \\ \varnothing: & \text{otherwise.} \end{cases} \tag{8}$$

with $o_{j,l}$ the maximal posterior probability (output of classifier l) and $o_{i,l}$ the second highest posterior probability. The empty set $\varnothing$ indicates that a case $\boldsymbol{o}_l = b_l(\boldsymbol{x}^{\neq \boldsymbol{v}_l})$ remains unclassified and is propagated to the next classifier $l+1$ in the cascade. By varying the thresholds $\boldsymbol{Z}$, one can control both how many features are typically needed to assign a class label to a case and the total performance of the cascade. Increasing the threshold values $\boldsymbol{z}_l$ implies that cases are more likely to be propagated to the next classifier $l+1$ in the cascade. Fig. 2 illustrates the principle of a cascade with three classifiers designed for a three-class problem. First, one feature is measured and provided as input to the first classifier. The classification rule either assigns a reliable class label to a case or propagates it to the next classifier after a second feature value has been measured.

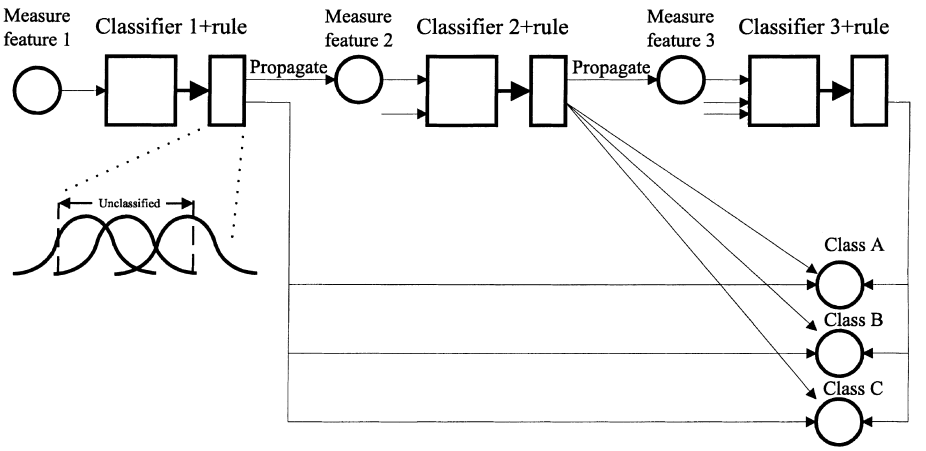

Fig. 2. Three classifiers are combined into a cascade. The first classifier assigns class labels to cases that can be classified reliably whereas other cases are propagated to the next classifier. The small rectangle next to each classifier represents the classification rule that is used corresponding to the threshold vector z. For each propagated case, an additional feature is measured.

3.6. Calibration of the cascade – equal acquisition costs

In the simple case where all features have equal acquisition costs, the relevant features in the cascade $\boldsymbol{C}$ have been pruned according to their marginal contribution. Denote with $\boldsymbol{X}$ the set of training cases and with $\boldsymbol{X}_l$ the subset of cases classified correctly by classifier l,

$$\boldsymbol{X}_l = \{\boldsymbol{x}^{\neq v_l} \in \omega_j | P^l(\omega_j|\boldsymbol{x}^{\neq v_l}) > P^l(\omega_i|\boldsymbol{x}^{\neq v_l}) \ \forall i \neq j\}. \tag{9}$$

Denote with $\boldsymbol{O}_l$ the corresponding posterior probabilities computed for the cases in $\boldsymbol{X}_l$. Denote with $\boldsymbol{X}_{n\setminus l}$ the subset of cases that are *correctly* classified by classifier n (based on all features) but classified *wrongly* by classifier l,

$$\boldsymbol{X}_{n\setminus l} = \boldsymbol{X}_n \setminus \boldsymbol{X}_l. \tag{10}$$

In the situation where one wants to obtain a cascade with the maximal correctness given equal feature acquisition costs, the classification rule $\boldsymbol{z}_l$ of each classifier l in the cascade should be calibrated such that all cases in $\boldsymbol{X}_{n\setminus l}$ are left unclassified by classifier l. When $\boldsymbol{O}_{n\setminus l}$ is the corresponding matrix containing the output vectors of classifier l, the threshold vector $\boldsymbol{z}_l$ can be assigned as follows:

$$z_{j,l} = \beta_{j,l} + \varepsilon_{j,l}, \tag{11}$$

with

$$\begin{aligned} &\beta_{j,l} = \max_r\{o_{r,j} - o_{r,u}\}, \quad o_{r,j} = \max_i(o_{r,i}),\\ &o_{r,u} = \max_{i\neq j}(o_{r,i}), \quad \boldsymbol{o}_r \in \boldsymbol{O}_{n\setminus l}, \end{aligned} \tag{12}$$

which indicates the maximal difference between the posterior probabilities of the correct, winning class j and its closest competitor. $\varepsilon_{j,l} > 0$ is a parameter related to the standard deviations $\sigma_{r,j}$ and $\sigma_{r,u}$ of the estimates $\boldsymbol{O}_{n\setminus l}$. Note that the rule (12) ensures that the threshold vector of each classifier in the cascade can be chosen independently of the threshold vectors of the subsequent classifiers.

3.7. Calibration of the cascade – varying acquisition costs

In the second situation characterized in Section 3.2, the acquisition costs vary between the features and a trade-off should be specified between a misclassification and the costs associated with measuring each feature. Assuming that the features have been pruned according to their

marginal utility γ_k defined in Eq. (6), the threshold vector of classifier l should be set such that

$$\frac{\mathrm{d}\gamma_k}{\mathrm{d}z} = 0 \Longleftrightarrow (\rho^{l+1}_{|z} - \rho^{l}_{|z})\, c(\overline{\omega}|\omega) = P^{l}_{z}(\varnothing) c(x_k) \tag{13}$$

for each classifier $l = 1, \ldots, n-1$, in the cascade. $\rho^{l}_{|z}$ denotes the correctness of classifier l in the cascade and $P^{l}_{z}(\varnothing)$ the probability that this classifier leaves a case unclassified, both for a given threshold vector $\boldsymbol{z}$. It is clear that the correctness of classifier l and the probability that it leaves cases unclassified both depend on how the preceding classifiers in the cascade are calibrated. So a total cost function needs to be minimized.

4. Experiments

Two cascades were built from neural networks trained for a difficult classification task: the prediction whether a patient will develop atrial fibrillation (AF) directly after cardiac surgery. The classification task is based on one continuous and 10 discrete features, see Table 1. These data were collected from 600 patients which had been monitored subsequent to cardiac surgery. Eight incomplete cases were removed so 592 cases were available for training and testing the networks, $P(\mathrm{AF}) = 0.53$. The cases were divided into a training set consisting of 474 cases and a test set consisting of 118 cases.

4.1. First experiment

In this experiment, we first investigated whether the potential influence measure was suited as an assessment criterion for feature selection according to a backward search scheme. The encoding scheme of the features is indicated in Table 1. Earlier experiments with these data had indicated that a topology with two hidden nodes resulted in the best generalization performance on a test set, so this network topology was kept during all experiments. Five neural networks with different initial weight configurations were trained for 3000 cycles with back-propagation, off-line learning. The learning rate was set to 0.001, momentum to 0.1. The average correctness on the test set was 0.7458 ($\pm$0.0053).

The potential influence ϕ_k, $k = 1, \ldots, 11$, was computed for each of the 11 features on the training set. The feature with the smallest potential influence was removed and five networks were trained on the reduced feature set. When more

Table 1
The 11 features available to the classifiers[a]

Feature	Description	Outcomes	Type	Coding
1	Whether the patient had AF when referred to the clinic	No, parox., chronic	Nom.	100, 010, 001
2	Whether post-operative beta-blocker was prescribed	Yes, no	Bin.	0, 1
3	Age	Real number	Cont.	Real number
4	Type of operation	CABG, valve, CABG + valve	Nom.	100, 010, 001
5	Contraindication for beta-blockers	Yes, no	Bin.	0, 1
6	Chronic obstructive pulmonic disease requiring broncho-dilating drugs	Yes, no	Bin.	0, 1
7	History of cong. heart failure	Yes, no	Bin.	0, 1
8	Ejection fraction < 40%	Yes, no	Bin.	0, 1
9	Post-operative heart failure requiring inotropic agents	Yes, no	Bin.	0, 1
10	Sinus bradycardia (<50 bpm) or sign. atrioventricular cond. delay	Yes, no	Bin.	0, 1
11	Peri- or post-operative sinus dysfunction or atriventricular cond. disturbanced	Yes, no	Bin.	0, 1

[a] Three types of features are distinguished: nominal, binary and continuous. In total 592 complete cases had been collected.

Table 2
Performance of five neural networks (two hidden nodes) on the training and test sets as a function of the number of features used

Correctness	2 features	3 features	4 features	5 features	6 features	7 features
Training set						
μ	0.7489	0.7506	0.7523	0.7566	0.7553	0.7637
σ	0.0000	0.0021	0.0017	0.0028	0.0071	0.0065
Test set						
μ	0.7288	0.7356	0.7407	0.7407	0.7322	0.7509
σ	0.0000	0.0083	0.0086	0.0115	0.0042	0.0068
Features	1, 2	1, 2, 4	1, 2, 4, 10	1, 2, 4, 10, 11	1–4, 10, 11	1–4, 9–11

than one feature had the same smallest potential influence, all these features were removed. This approach led to inferior classifiers as feature 4 was removed rather early in the backward search procedure (when seven features remained). Without this feature, too many cases become misclassified. So potential influence is only suited as an assessment criterion when completely redundant features can be identified, $\phi_k = 0$.

Pruning features using a backward search was continued using the marginal contribution as the assessment criterion. The correctness on the training and test sets is shown in Table 2. Pruning was stopped when two features remained. The performance on the test set peaked when four and five features were used, whereas seven features resulted in a slightly higher performance than the configurations with four and five features.

4.2. Second experiment

In the second experiment, we investigated the possibility of building a cascade of neural networks. Two cascades were built: one consisting of 2 networks (2 and 3 features) and another consisting of 3 networks (2, 3 and 5 features). The reason that a network with 4 features was not used in the second cascade was that no single network among the 5 networks trained with 4 features had a performance (on the training set) that exceeded the best network using 3 features.

Table 3
Correctness (fraction of correctly classified cases), coverage (fraction of classified cases) and number of actually classified cases for each classifier in the two cascades computed on the training set (474 cases)[a]

Training set	Network 1 (2 f.)	Network 2 (3 f.)	Network 3 (5 f.)
Correctness	0.7583	0.6522	
Coverage	0.9515	1.0000	
# cl. cases	451	23	
Avg. feature costs	2.05		
Total correctness	0.7532		
Correctness	0.7921	0.7746	0.6500
Coverage	0.6392	0.4152	1.0000
# cl. cases	303	71	100
Avg. feature costs	2.78		
Total correctness	0.7595		

[a] The first cascade contains two networks, one that uses 2 features as input, the other network 3 features. The second cascade contains three networks, one that uses 2 features as input, the other network 3 features and the last 5 features. For each cascade, the average number of features measured per case and the total correctness are computed.

Table 4
Correctness, coverage and number of actually classified cases for each classifier in the two cascades computed on the test set (118 cases)[a]

Test set	Network 1 (2 f.)	Network 2 (3 f.)	Network 3 (5 f.)
Correctness	0.7615	0.5556	
Coverage	0.9237	1.0000	
# cl. cases	109	9	
Avg. feature costs	2.08		
Total correctness	0.7458		
Correctness	0.7595	0.9333	0.6667
Coverage	0.6695	0.7692	1.0000
# cl. cases	79	30	9
Avg. feature costs	2.48		
Total correctness	0.7966		

[a] For each cascade, the average number of features measured per case and the total correctness are computed.

The threshold vector z_1 in the first cascade was calibrated such that no case, which was classified correctly by the network with 3 features, was misclassified by the first network with 2 features. Table 3 shows the correctness (fraction of classified cases assigned a correct class label), coverage (fraction of cases assigned a class label, see Egmont-Petersen et al., 1994) per network and the average number of features measured in the cascaded classifier computed on the 474 cases in the training set. Table 4 contains the same statistics but computed on the 118 test cases.

In the second cascade, the two threshold vectors, z_1 and z_2, were calibrated such that no case, which was classified correctly by the network with 5 features, was misclassified by the two preceding networks (using 2 and 3 features). Tables 3 and 4 show the performance of each of the classifiers, the average number of features measured and the correctness of the cascade computed on the training and test sets.

5. Discussion and conclusion

The experiment with the cascades indicates that the measurement of extra features results in a higher total correctness, while the average number of features measured may be kept low. So for *sequential* classification problems, building a cascade is worthwhile. The extent to which the classification task is sequential determines the decrease in average acquisition costs that can be obtained when building a cascade.

In general, prototypical cases can often be classified reliably using only a few feature values, whereas borderline cases can only be classified when most of the n feature values are known. Consequently, prototypical cases can be assigned a reliable class label early in the cascade, whereas borderline cases are propagated further requiring more features to be measured.

For classification problems where the performance of the classifier peaks, adding particular features leads to a decrease in correctness. Such features should not be used in a cascade. So a thorough feature selection procedure is a prerequisite for building a cascade. A consequence is that several neural networks need to be trained and should be treated as an ensemble. Moreover, the certainty factor $\varepsilon_{j,l}$ can only be chosen optimally when the standard deviation $\sigma_{r,j}$ of the output vectors of the classifiers is estimated from such an ensemble. In the classification problem considered here, the prediction of atrial fibrillation, a peak occurs around 4–5 features. Building a cascade with 7, 8–11 features would require larger training and test sets.

The applicability of the cascade concept depends also on the domain. In digital image pro-

cessing, one is typically interested in lowering the computational complexity by removing superfluous features. When the performance significantly drops, however, the pruning procedure will be stopped. For medical classification tasks such as diagnosis, the acquisition of certain features may be undesired (e.g. performing a brain biopsy) and should be postponed as long as possible. Building a cascade under such circumstances has only briefly been considered here and is the subject of future research.

Discussion

Duin: You said that this approach is different from decision trees. But once you consider the output of classifiers as new features, then I think it is very similar to building a decision tree.

Egmont-Petersen: That is true. If you choose that option, you are right. I have not experimented with using both the features and the output of the previous classifier as input to the subsequent classifier. I only propagated the features that were already known. This is certainly an interesting extension to the approach.

Pudil: I just would like to comment that your approach seems to have very much in common with what we presented at the ICPR in The Hague in 1992. (*Note of the editors*: *P. Pudil, J. Novovicová, S. Blahá, J. Kittler. Multistage Pattern Recognition with Reject Option. Proceedings of the 11th ICPR, IEEE Computer Society Press, vol. II, Los Alamitos, 1992, pp. 92–95*). We presented the idea of a hierarchical feature selection scheme or a set of feature selection classifiers. That was also based on the idea of acquisition cost of features. We derived an average cost function for decision making, taking into account the acquisition cost. For example, we started with cheaper features to classify simple cases, and if such cases were not classified unambiguously, they were sent to the next classifier, and so on. We found that such a scheme yields in most cases much lower average decision risk than using a one-level scheme. The idea was to process simple cases with fewer and cheaper features and processing only the very complicated patterns or cases with more advanced, more informative, but more costly features.

Egmont-Petersen: So that is a positive result, I understand, and it supports the proposed method.

Inza: Just a question: is your proposed measure monotonic as a function of the number of features?

Egmont-Petersen: That is a very good question. It depends on whether there is a peak. This again depends on the specific combination of the classifier, the domain and the number of cases. So, for any given problem, I cannot guarantee that there will be no peak. On the contrary, I would say that for many domains, depending on the size of the feature set and the number of cases, there will be a peak. So that will have to be investigated to begin with.

Gelsema: In feature selection there is always the problem of optimality. Now I do not know if in this case you can define optimality, and my question is: did you define optimality and can you then guarantee that your subset is the optimal subset?

Egmont-Petersen: First of all, when you are looking for the optimal subset of features, if you run into the peaking phenomenon, the only way to find the optimal feature subset is by exhaustive search. In this work I have presented formulas which define optimality as the optimal trade-off, the marginal utility as I call it. This gives the optimal trade-off between feature acquisition costs and the cost of misclassifying cases. I present some formulas on how to find the equilibrium between these costs by setting the thresholds.

References

Duda, R.O., Hart, P.E., 1973. Pattern Classification and Scene Analysis. Wiley, New York.

Egmont-Petersen, M., 1996. Specification and Assessment of Methods Supporting the Development of Neural Networks in Medicine. Shaker Publishing, Maastricht.

Egmont-Petersen, M., Pelikan, E., 1999. Detection of bone tumours in radiographs using neural networks. Pattern Analysis and Applications 2 (2), 172–183.

Egmont-Petersen, M., Talmon, J.L., Brender, J., McNair, P., 1994. On the quality of neural net classifiers. Artificial Intelligence in Medicine 6 (5), 359–381.

Egmont-Petersen, M., Talmon, J.L., Hasman, A., Ambergen, A.W., 1998. Assessing the importance of features for multi-layer perceptrons. Neural Networks 11 (4), 623–635.

Hamamoto, Y., Uchimura, S., Tomita, S., 1996. On the behavior of artificial neural network classifiers in high-dimensional spaces. IEEE Trans. Pattern Anal. Mach. Intell. 18 (5), 571–574.

Pudil, P., Novoviová, J., Bláha, S., Kittler, J., 1992. Multistage pattern recognition with reject option. In: Proc. 11th IAPR Internat. Conf. on Pattern Recognition, The Hague. IEEE Computer Society Press, Los Alamos, pp. 92–95.

Quinlan, J.R., 1993. Comparing connectionist and symbolic learning methods. In: Hanson, S., Drastal, G., Rivest, R. (Eds.). Computational Learning Theory and Natural Learning Systems: Constraints and Prospects. MIT Press, Cambridge.

Tian, B., Shaikh, M.A., Sadjadi, M.R., Haar, T.H., Reinke, D.L., 1999. A study of cloud classification with neural networks using spectral and textural features. IEEE Trans. Neural Networks 10 (1), 138–151.