On Feature Selection with Measurement Cost and Grouped Features

Pavel Paclík[1], Robert P.W. Duin[1],
Geert M.P. van Kempen[2], and Reinhard Kohlus[2]

[1] Pattern Recognition Group, Delft University of Technology
The Netherlands
{pavel,duin}@ph.tn.tudelft.nl
[2] Unilever R&D Vlaardingen, The Netherlands
Geert-van.Kempen@unilever.com
Reinhard.Kohlus@unilever.com

Abstract. Feature selection is an important tool reducing necessary feature acquisition time in some applications. Standard methods, proposed in the literature, do not cope with the measurement cost issue. Including the measurement cost into the feature selection process is difficult when features are grouped together due to the implementation. If one feature from a group is requested, all others are available for zero additional measurement cost. In the paper, we investigate two approaches how to use the measurement cost and feature grouping in the selection process. We show, that employing grouping improves the performance significantly for low measurement costs. We discuss an application where limiting the computation time is a very important topic: the segmentation of backscatter images in product analysis.

1 Introduction

Feature selection is usually used to choose a feature subset with the best possible performance [2,3]. The acquisition cost of selected features is, however, also an important issue in some applications. We can mention, for example, medical diagnosis or texture segmentation. In this paper, we are interested in cases where well-performing cheap features are preferred over the expensive ones delivering just slightly better results.

Due to the implementation efficiency, features are often produced in groups. The computation of time-consuming intermediate results is performed just once and then used for the generation of a number of different features. Examples may be Fourier descriptors or various texture features. If the traditional feature selection technique is used in such a case, the resulting feature subset will often require unnecessarily long acquisition time when classifying of new objects. In this paper, we discuss a strategy how to include the information about the feature grouping into the feature selection process and thereby save the time.

An application that enabled our interest in the feature selection with measurement cost is the segmentation of backscatter images (BSE) in the analysis

T. Caelli et al. (Eds.): SSPR&SPR 2002, LNCS 2396, pp. 461–469, 2002.

of laundry detergents [4]. Let us use it as an example of a problem where the speeding-up of the feature computation is an important design issue. The development of laundry detergents is based on structural analysis of BSE images. For each powder type, the batch of BSE images is acquired, segmented and analyzed. Image segmentation is performed by supervised statistical pattern recognition algorithm using a number of mainly texture features [5]. Feature selection is run on a single training image. The batch of BSE images from the same powder formulation is then segmented by a trained texture classifier. An important point is that the feature selection is performed for each new batch of images due to variable magnification and type of detergent structures to be labeled. Feature acquisition is computationally intensive problem as the image pixels of high-resolution images are treated as individual data samples. Taking into account also the number of processed images within a batch, the feature selection method should optimize both the performance and the feature computation time. From the implementation point of view, features form several groups. Our intention is to use feature grouping in the feature selection process to find a time-effective feature set.

In the next section, we explain two strategies how the feature grouping may be employed in the feature selection. In the section 3, we discuss experiments on two different problems: handwritten digit recognition and backscatter image segmentation. Finally, in the last section, we give conclusions.

2 Feature Selection with Grouped Features

Feature selection algorithm searches for the best subset of d features from a complete set of D measurements, $d < D$. Several searching strategies have been proposed in the literature [2,3]. In the following, we use the sequential forward feature selection algorithm (SFS). Selection of features is based on a criterion function. In the paper, we use performance of a classifier on an evaluation set as a criterion. It predicts the performance degradation caused by the use of weak features in high dimensionalities (curse of dimensionality).

Standard feature selection algorithms do not take into account measurement cost. Therefore, expensive features may be selected while a number of weaker features is available at low cost. Measurement cost may be combined into the selection criterion in several different ways. In this paper, we consider a criterion C of the following form:

$$C = \frac{\Delta P}{\Delta T}. \tag{1}$$

Here, ΔP stands for the increase of performance and ΔT denotes the increase of measurement cost between two algorithm steps. This criterion favors cheap features offering a small performance gain before better but expensive ones. If linear weighting of performance and measurement cost is of interest, different criterion might be a better choice.

In reality, implementation often defines grouping of features. Group G of N features is computed at once. If one feature from the group is used, all others are available for zero additional measurement cost. If time optimization is of interest, adding descriptive features with zero measurement cost should be preferred. Unfortunately, zero increase of the measurement cost poses a problem for the selection algorithm using criterion (1).

We propose to change the selection strategy and choose the features on the *per-group* basis. It means, that the feature selection algorithm runs at two levels. At the higher level, it operates over feature groups. For each group, a convenient feature subset is found. The performance of the selected subset in the group is used to choose the best group. We have been investigating two variants of this approach.

2.1 Group-Wise Forward Selection (GFS)

In this method, forward feature selection is run for each group. A group is judged based on the performance of its *all* features. For the group with the best score, all the features are included to the list of selected features. The method, which is fast and easy to implement, is appropriate in cases where including all the features from the group does not dramatically decrease the system performance. In the following algorithm, function `getcost(`subset`)` returns relative measurement cost of the feature subset and `getperf(`data,subset`)` returns subset performance.

2.2 Group-Wise Nested Forward Selection (GNFS)

The main idea of this method is to use the best feature subset in the group instead of all the group's features. In order to identify such a subset, nested feature selection search is launched within each group. The group is judged on the basis of its best feature subset. Features that were not selected in one step may be used later for a zero additional measurement cost.
GNFS algorithm keeps track of group-specific information (structure *group*, lines 6-11). Newly computed features are judged by the criterion (1) while features from already computed groups are judged solely by their performance. If a subset of already computed and therefore cheap features may be found, which improves the performance, it is used preferably to features from a new group offering a bigger performance gain. This decision is made on the line 18 of the GNFS algorithm and its implementation was omitted for the sake of simplicity.
If just single feature groups are present, both proposed algorithms perform sequential forward selection with criterion (1).

3 Experiments

3.1 Handwritten Digit Recognition

In the first experiment, we use the proposed methods on the handwritten digit `mfeat` dataset from [1]. The dataset contains 10 digit classes with 200 samples per class and six different feature sets (649 features). In order to lower the computational requirements in this illustrative example, we have reduced the number of features in all of the six feature sets. The dataset used in this experiment contains 78 features. The set with 100 samples per class was used for training; the other 100 samples per class were used as the evaluation set for the feature selection criterion (error of the linear discriminant classifier assuming normal densities).

Experimental results are presented in Figure 1. Performance of selected feature subset on the evaluation set is estimated as a function of the measurement cost. The measurement cost is expressed on the relative scale $\langle 0, 1.0 \rangle$, where 1.0 corresponds to the computational time of all the features. Because the computational time of individual feature groups is not known for this dataset, we assume equal measurement cost for all the features. Measurement cost of particular group is then a sum of measurement costs for the group's features.

The solid line in the graph represents results of the forward feature selection algorithm not using the feature grouping. The dashed line with cross markers corresponds to GFS, and dash-dotted line with circles to GNFS algorithm.

Points on the curves denote steps of the corresponding feature selection algorithms. While one step represents one added feature for SFS method, for GFS is that adding of all and for GNFS adding of the subset of the group's fea-

Algorithm 1 Group-wise Forward Selection (GFS)

```
input: data, features $F = \{1, ..., D\}$, feature groups $G = \{1, ..., N\}$
$C_{max} = 0$; $F_{best} = \{\}$;                  // best criterion and subset found
$G_{sel} = \{\}$; $F_{sel} = \{\}$;                // selected groups, selected features
currperf = 0;                                      // performance of the current subset $F_{sel}$
currcost = 0;                                      // meas.cost of the current subset $F_{sel}$
while length($G_{sel}$)$<N$
    $C = \{\}$; perf = $\{\}$;                     // criteria values, performance values
    for i=1 to length($G$)
        $A \leftarrow$ all features from group $G(i)$;
        mincr =getcost($\{F_{sel} \cup A\}$)$-$currcost;      // measurement cost increase
        perf($i$) =getperf(data,$\{F_{sel} \cup A\}$);
        $C(i)$ = (perf($i$) $-$ currperf)/mincr;              // criterion
    end
    $m$=max(C);                                    // maximum criterion value
    $imax$=argmax(C);                              // index of the best group
    $F_{sel} \leftarrow$ features from group $G(imax)$;
    currperf = perf($imax$);
    currcost =getcost($F_{sel}$);
    $G_{sel} \leftarrow G(imax)$;                  // add group to list
    $G(imax) = \emptyset$;                         // remove group from list $G$
    if $m > C_{max}$
        $C_{max} = m$; $F_{best} = F_{sel}$;       // adjust the best achieved criterion and subset
    end
end
output: the best subset found $F_{best}$
```

tures. Note also the vertical curve segment on the solid line which is caused by adding features with a zero measurement cost (they come from already computed groups).

It can be seen, that both methods using the feature grouping reach a very good result (0.028) already for one third of the computation time. The standard method achieves the similar performance for 48% of the measurement cost.

The lower graph in the Figure 1 presents the number of used features as a function of the relative measurement cost for all three methods. It can be seen, that methods using feature grouping perform better than the standard selection due to larger number of employed features at the same measurement cost.

3.2 Backscatter Image Segmentation

In the second experiment, we apply proposed methods in order to speed-up the feature acquisition process in the backscatter image segmentation. For the sake of feature selection, a dataset with 3000 samples, three classes, and 95 features

Algorithm 2 Group-wise Nested Forward Selection (GNFS)

```
input: data, features $F = \{1, ..., D\}$, feature groups $G = \{1, ..., N\}$
$C_{max} = 0$; $F_{best} = \{\}$;                 // best criterion and subset found
$F_{sel} = \{\}$;                                 // selected features
currperf = 0;                                     // performance of the current subset $F_{sel}$
currcost = 0;                                     // measurement cost of the current subset $F_{sel}$
for i=1 to $N$
    group($i$).perf = 0;                          // set-up auxiliary group structure
    group($i$).computed = 0;                      // is this group already computed?
    group($i$).$F_{sel}$ = {};                    // best found subset in the group
    group($i$).$F$ ← features from group $G(i)$;
end
while length($F_{sel}$)<$D$
    for i=1 to $N$                                // perform nested search for each group
        group($i$).perf = 0;
        $A$ = group($i$).$F$;                     // assign not-yet-used features
        group($i$) ← find best subset and its performance on $A$;
    end
    $m, imax$ ← choose the best subset;
    $F_{sel}$ ← group($imax$).$F_{sel}$                           // added features
    group($imax$).$F$ = group($imax$).$F$ \ group($imax$).$F_{sel}$;        // remove selected
    group($imax$).$computed$ = 1;                                // toggle group as computed
    currperf = group($imax$).perf;
    currcost =getcost($F_{sel}$);
    if $m > C_{max}$
        $C_{max} = m$; $F_{best} = F_{sel}$;     // adjust the best achieved criterion and subset
    end
end
output: the best subset found $F_{best}$
```

was computed from a BSE image. Six different types of features were used: intensity features, cooccurrence matrices (CM), gray-level differences (SGLD), local binary patterns (LBP), features based on Discrete Cosine Transform (DCT), and Gabor filters. More details regarding feature types and segmentation algorithm can be found in [5].

The Table 2 summarizes the actual feature grouping defined by the used implementation. It follows from the table, that each of eight DCTs and 24 Gabor filters is computed apart forming a separate group. The last column in the table indicates a relative cost to compute the group (1.0 is the total cost using all groups).

The experimental results for four different backscatter images are presented in Figure 3. A complete dataset with 95 features was computed for each image. Then, three feature selection methods were run (standard forward selection not using feature grouping and two presented methods employing grouping information). Once again, the error of the linear discriminant classifier assuming normal densities on the independent evaluation set was used as the criterion. Evaluation

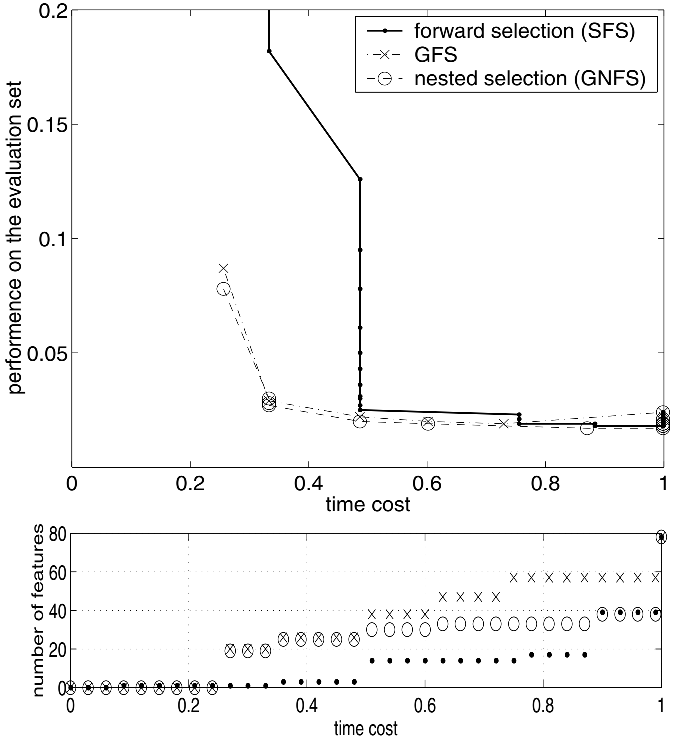

Fig. 1. Performance as a function of the measurement cost for handwritten digit dataset (upper plot). Number of features as a function of the measurement cost (lower plot)

group number	*feature type*	*features per group*	*group cost*
1	intensity	4	0.0091
2	CM	4	0.0422
3	SGLD	4	0.0118
4	LBP1	17	0.0694
5	LBP2	25	0.1009
6	LBP3	9	0.0434
7-14	DCT filters	1	0.0349
15-38	Gabor filters	1	0.0185

Fig. 2. Feature groups in the backscatter segmentation experiments

set consists of different 3000 samples from the same BSE image. All the three lines end up in the same point (performance of the complete feature set).

Maximum performance for all methods is summarized in the Table 4. It appears, that the best achieved performance is similar in all cases. Proposed methods utilizing feature grouping lower the measurement cost of feature computation for all but the last image. Further examination of performance-cost curves in Figure 3 suggests possible better choice of operating points with lower measurement costs.

It is interesting to note areas where the standard feature selection algorithm finds better solutions than both proposed methods (images 2 and 3 in Figure 3). We think, that the reason is in the fine-grained approach of the standard algorithm. Both proposed algorithms outperform standard forward feature selection for low measurement costs which is our area of interest. In general, nested feature selection (GNFS) works better than adding all the group's features (GFS) but is computationally more intensive.

4 Conclusions

We investigate ways how the information about feature grouping may be used in the feature selection process for finding well-performing feature subset with low measurement cost. The problem arises in many real applications where the feature acquisition cost is of importance and feature grouping is defined by the implementation.

We show, that it is beneficial to perform feature selection on the per-group basis. Different strategies may be then chosen to select appropriate feature subset within the groups. We have investigated two such approaches – adding all features from the group (GFS) and Group-wise nested feature selection (GNFS).

It follows from our experiments on handwritten digits and backscatter images, that proposed methods outperform standard feature selection algorithm in the low measurement cost area. We also conclude, that using nested feature selection is better strategy than adding all group's features but it is computationally more intensive.

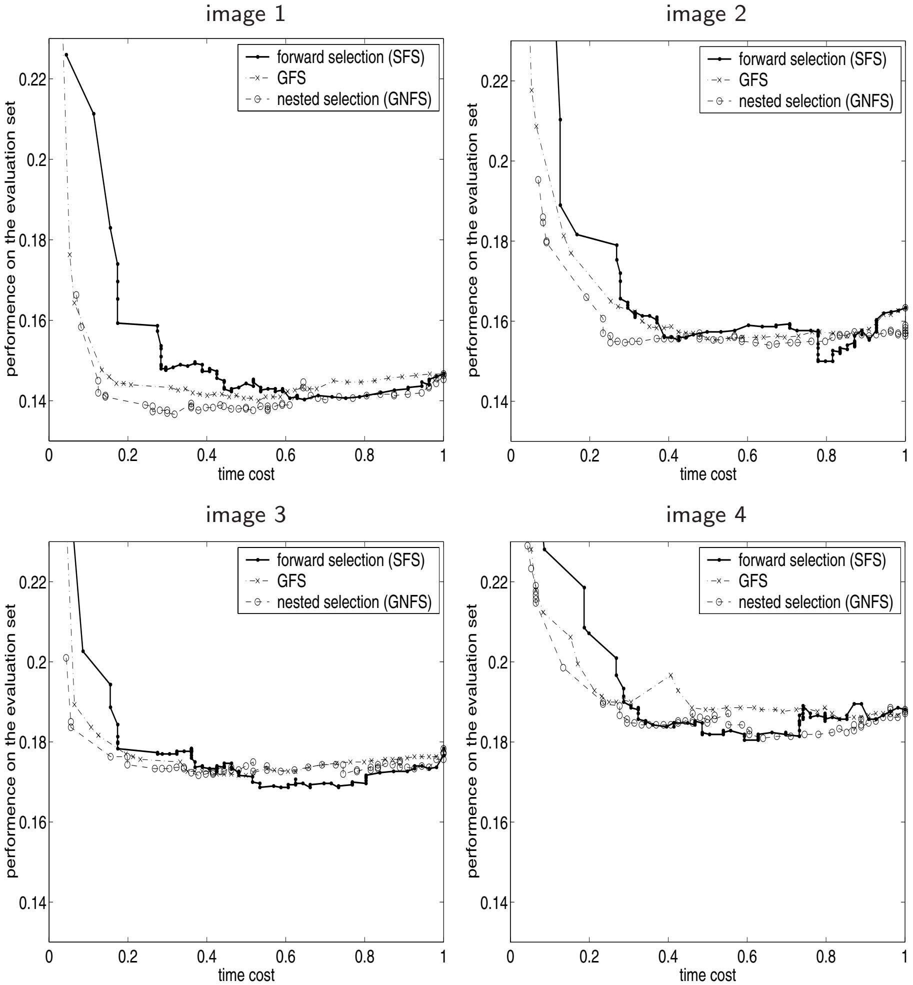

Fig. 3. Performance as a function of the measurement cost for backscatter segmentation experiment

image	forward selection (SFS)	GFS	GNFS
1	0.140 at 0.65 (77)	0.142 at 0.30 (59)	0.138 at 0.32 (50)
2	0.150 at 0.78 (57)	0.158 at 0.56 (77)	0.154 at 0.65 (69)
3	0.169 at 0.53 (60)	0.174 at 0.28 (49)	0.172 at 0.38 (48)
4	0.181 at 0.59 (48)	0.181 at 0.83 (63)	0.181 at 0.64 (71)

Fig. 4. Best performances of feature subsets in backscatter segmentation experiment. The first number is the best performance, the second is corresponding measurement cost and the number if parentheses is the feature count

Presented methods are based on simple forward feature selection algorithm. It is possible to replace forward selection by more powerful methods like floating search [6]. Computation time of the feature selection may, however, increase considerably what may be not acceptable in some applications.

References

1. C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998. `http://www.ics.uci.edu/ |mlearn/MLRepository.html`. 463
2. F. Ferri, P. Pudil, M. Hatef, and J. Kittler. Comparative study of techniques for large-scale feature selection, 1994. 461, 462
3. Anil Jain and Douglas Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE Trans. Pattern Analysis and Machine Inteligence*, 19(2):153–158, February 1997. 461, 462
4. Pavel Paclík, Robert P.W. Duin, and Geert M.P. van Kempen. Multi-spectral Image Segmentation Algorithm Combining Spatial and Spectral Information. In *Proceedings of SCIA 2001 conference*, pages 230–235, 2001. 462
5. Pavel Paclík, Robert P.W. Duin, Geert M.P. van Kempen, and Reinhard Kohlus. Supervised segmentation of backscatter images for product analysis. accepted for International Conference on Pattern Recognition, ICPR2002, Quebec City, Canada, August 11-15, 2002. 462, 466
6. P. Pudil, J. Novovičová, and Kittler J. Floating search methods in feature selection. *Pattern Recognition Letters*, 15:1119–1125, 1994. 469