P. Perner, Improving the Accuracy of Decision Tree Induction by Feature Pre-Selection, Applied Artificial Intelligence 2001, vol. 15, No. 8, p. 747-760.

Improving the Accuracy of Decision Tree Induction by Feature Pre-Selection

Petra Perner
Institute of Computer Vision and Applied Computer Sciences
Arno-Nitzsche-Str. 45,04277 Leipzig
e-mail: ibaiperner@aol.com http://www.ibai-research.de

Running Head: Feature Subset Selection

Corresponding mail Address: Petra Perner
Institute of Computer Vision and Applied Computer Sciences
Arno-Nitzsche-Str. 45
04277 Leipzig, Germany

P. Perner, Improving the Accuracy of Decision Tree Induction by Feature Pre-Selection, Applied Artificial Intelligence 2001, vol. 15, No. 8, p. 747-760.

Abstract. Selecting the right set of features for classification is one of the most important problems in designing a good classifier. Decision tree induction algorithms such as C4.5 have incorporated in their learning phase an automatic feature selection strategy while some other statistical classification algorithm require the feature subset to be selected in a preprocessing phase. It is well know that correlated and irrelevant features may degrade the performance of the C4.5 algorithm. In our study, we evaluated the influence of feature pre-selection on the prediction accuracy of C4.5 using a real-world data set. We observed that accuracy of the C4.5 classifier could be improved with an appropriate feature pre-selection phase for the learning algorithm. Beyond that, the number of features used for classification can be reduced which is important for image interpretation tasks since feature calculation is a time-consuming process.

1 Introduction

Selecting the right set of features for classification is one of the most important problems in designing a good classifier. Very often we don’t know a-priori what the relevant features are for a particular classification task. One popular approach to address this issue is to collect as many features as we can prior to the learning and data-modeling phase. However, irrelevant or correlated features, if present, may degrade the performance of the classifier. In addition, large feature spaces can sometimes result in overly complex classification models that may not be easy to interpret.

In the emerging area of data mining applications, users of data mining tools are faced with the problem of data sets that are comprised of large numbers of features and instances. Such kinds of data sets are not easy to handle for mining. The mining process can be made easier to perform by focussing on a subset of relevant features while ignoring the other ones. In the feature subset selection problem, a learning algorithm is faced with the problem of selecting some subset of features upon which to focus its attention.

In this paper, we present our study on features subset selection and classification with the C4.5 algorithm. In Section 2, we briefly describe the criteria used for feature selection and the feature selection methods. Although, C4.5 has a feature selection strategy included in its learning performance, it has been observed that this strategy is not optimal. Correlated and irrelevant attributes may degrade the performance of the induced classifier. Therefore, we use feature subset selection prior to the learning phase. Besides the feature selection strategy used by C4.5 we describe two other feature subset selection algorithms in Section 2: the contextual merit algorithm (CM) (Hong, 1999) and the sequential floating forward selection algorithm (SFFS) (Pudil et al., 1994).

For our experiments, we used a real data set that includes features extracted from x-ray images which describe defects in a welding seam. It is usually unclear in these applications what the right features are. Therefore, most analyses begin with as many features as one extracts from the images. This process as well as the images is described in Section 3.

In Section 4, we describe our results. We show that the prediction accuracy of the C4.5 classifier will improve when provided with a pre-selected feature subset. The best result can be shown for the SFFS algorithm. The results show that the feature subsets created by both feature subset selection algorithm and the feature subset normally extracted by C4.5 have many features in common. However, the C4.5 selects some features that are never selected by the feature subset selection algorithm. We hypothesize that irrelevant features are weeded out by the feature selection algorithm while they get selected by the C4.5 algorithm. A comparison of the feature ranking done by the feature selection algorithm with the ranking of the features done by C4.5 for the first 10 features used by C4.5 shows that there is a big difference. Finally, our experiments also indicate that model complexity does not significantly change for the better or worse when pre-selecting features with feature selection.

2 Feature Subset Selection Algorithms

Following Jain et al. (Jain and Zonker, 1997), we can describe feature subset selection as follow: Let *Y* be the original set of features, with cardinality *n*. Let *d* represent the desired number of features in the selected subset *X*, $X \subseteq Y$. Let the feature selection criterion function for the set *X* be represented by *J(X).* Without any loss of generality, let us consider a higher value of *J* to indicate a better feature subset. Formally, the problem of feature selection is to find a subset $X \subseteq Y$ *such that* $/X/=d$ *and* $J(X) = \max_{Z \subseteq Y, |Z|=d} J(Z)$. An exhaustive approach to this problem would require examining all $\binom{n}{d}$ possible *d*-subsets of the feature set *Y*.

Feature selection algorithm differ in the search strategy, the feature selection criteria, the way they add or delete an individual feature or a subset of features at each round of feature selection and the overall model for feature selection.

According to the quality criteria (Nadler and Smyth, 1993) for feature selection, the model for feature selection can be distinguished into the filter model and the wrapper model (Cover, 1977), (Kohavi and John, 1998). The wrapper model attempts to identify the best feature subset for use with a particular algorithm, while the filter approach attempts to assess the merits of features from the data alone. The inclusion of the particular classifier into the feature selection process makes the wrapper approach more computationally expensive and the resulting feature subset will only be appropriate for the used classifier while the filter approach is classifier independent. However, both models require a search strategy that should come close to optimal. Various search strategies have been developed in order to reduce the computation time.

There have been developed different filter approaches over time in traditional pattern recognition and artificial intelligence. Two well-known algorithms that have been developed within the artificial intelligence community are FOCUS (Almuallim and Diettrich, 1994) and RELIEF (Kira and Rendell, 1992). The FOCUS algorithm starts with an empty feature set and carries out exhaustive search until it finds a minimal combination of features. It works on binary, noise-free data. RELIEF algorithm assigns a relevance weight to each feature, which is meant to denote the relevance of the feature to the target concept. RELIEF samples instances randomly from the training set and updated the relevance values based on the difference between the selected instance and the two nearest instances of the same and opposite classes. Lui et al. (Lui and Setiono, 1996) proposed a probabilistic approach to the search problem of the optimal set of features based on the Las Vegas algorithm. Koller et al. (Koller and Sahami, 1996) proposed an approach based on using cross-entropy to minimize the amount of predictive information loss during feature elimination. It starts with the full feature set and eliminates feature according to the selection criteria. It leads to suboptimal results. Both approaches should be easy to implement and they are efficient in handling large data sets.

An overview of feature subset selection algorithm from the pattern recognition side is given in Jain et al. (Jain and Zonker, 1997). Beyond the exhaustive search, branch-and-bound feature selection can be used to find the optimal subset of features much more quickly than exhaustive search. One drawback is that the branch-and-bound procedure requires the feature selection criteria to be monotone. A suboptimal feature selection method that has been shown results that come close to optimal is the SFFS algorithm.

Beyond that, some common learning algorithm have built in feature selection, for example, C4.5. Clustering of features and selecting groups of features that are most dissimilar is also a common approach for feature subset selection (Perner, 2000).

We have selected for our study three suboptimal approaches: C4.5 algorithm, the CM algorithm that has some similarity with RELIEF algorithm, and the SFFS algorithm.

The feature selection in C4.5 may be viewed as a filter approach, just as the CM algorithm and the SFFS algorithm. The CM algorithm selects features according to their "obligation" to the class discrimination in the context of other features. In opposition to that, the SFFS algorithm selects features according to their statistical correlation between each feature and the class. Besides that, both algorithms differ in the search strategy. While the SFFS algorithm is a traditional technique used in pattern recognition, the CM algorithm is a new algorithm developed by the data mining community.

2.1 Feature Selection done by Decision Tree Induction

Determining the relative importance of a feature is one of the basic tasks during decision tree generation. The most often used criteria for feature selection is information theoretic based such as the Shannon entropy measure I for a data set. If we subdivide a data set using values of an attribute as separators, we obtain a number of subsets. For each of these subsets we can compute the information value I_i. I_i will be smaller than I, and the difference (I-I_i) is a measure of how well the attribute has discriminated between different classes. The attribute that maximizes this difference is selected.

The measure can also be viewed as a class separability measure. The main drawback of the entropy measure is its sensitivity to the number of attributes values (White and Lui, 1994). Therefore C4.5 uses the gain ratio. However, this measure suffers the drawback that it may choose attributes with very low information content (Lopez de Mantaras, 1991).

C4.5 (Quinlan, 1993) uses a univariate feature selection strategy. At each level of the tree building process only one attribute, the attribute with the highest values for the selection criteria, is picked out of the set of all attributes. Afterwards the sample set is split into sub-sample sets according to the values of this attribute and the whole procedure is recursively repeated until only samples from one class are in the remaining sample set or until the remaining sample set has no discrimination power anymore and the tree building process stops.

As we can see feature selection is only done at the root node over the entire decision space. After this level, the sample set is split into sub-samples and only the most important feature in the remaining sub-sample set is selected. Geometrically it means that the search for good features is only done in orthogonal decision subspaces, which might not represent the real distributions, beginning after the root node. Thus, unlike statistical feature search strategies (Fukunaga, 1990) this approach is not driven by the evaluation measure for the combinatorial feature subset; it is only driven by the best single feature. This might not lead to an optimal feature subset in terms of classification accuracy.

Decision tree users and researchers have recognized the impact of applying a full set of features to a decision tree building process versus applying only a judiciously chosen subset. It is often the case that the latter produces decision trees with lower classification errors, particularly when the subset has been chosen by a domain expert. Our experiments were intended to evaluate the effect of using multivariate feature selection methods as pre-selection steps to a decision tree building process.

2.2 Contextual Merit Algorithm

For our experiment, we used the contextual merit (CM) algorithm (Hong, 1996). This algorithm employs a merit function based upon weighted distances between examples which takes into account complete feature correlation's to the instance class. The motivation underlying this approach was to weight features based upon how well they discriminate instances that are close to each other in the Euclidean space and yet belong to different classes. By focusing upon these nearest instances, the context of other attributes is automatically taken into account.

To compute contextual merit, the distance d^{k}_{rs} between values z_{kr} and z_{ks} taken by feature k for examples r and s is used as a basis. For symbolic features, the inter-example distance is 0 if $z_{kr} = z_{ks}$, and 1 otherwise. For numerical features, the inter-example distance is $\min\left(\frac{z_{kr} - z_{ks}}{t_k}, 1\right)$, where t_k is a threshold for feature k (usually 1/ 2 of the magnitude of range of the feature). The total distance between examples r and s is $D_{rs} = \sum_{k=1}^{N_f} d^{k}_{rs}$ with N_f the total number of features and the contextual merit for a feature f is $M_f = \sum_{r=1}^{N} \sum_{s \in \bar{C}(r)} w^{f}_{rs} d^{f}_{rs}$, where N is the total number of examples, $\overline{C(r)}$ is the set of examples not in the same class as examples r, and w^{f}_{rs} is a weight function chosen so that examples that are close together are given greater influence in determining the merit of each feature. In practice, it has been observed that $w_{ij} = \frac{1}{D^{2}_{rs}}$ if s is one of k nearest neighbors to r, and 0 otherwise, provides robust behavior as a weight function. Additionally, using $\log_2 \left|\overline{C(r)}\right|$ as the value for k has also exhibited robust behavior. This approach to computing and ordering features by their merits has been observed to be very robust, across a wide range of examples.

2.3 Floating Search Method

The feature subset selection algorithm described in the former chapter performs in the first step a greedy search over all examples and it focuses afterwards on the computation of the $\log_2 \left|\overline{C(r)}\right|$ nearest examples based upon an Euclidean space. The first step is a very time-consuming process and the question arises if second step of the algorithm, where the real merits are calculated, will lead to a near optimal solution.

Various other search strategies have been developed to find the subset of features optimizing an adopted criterion. The well-known Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS) are step-optimal only since the best (the worst) feature is always added (discarded) in SFS and SBS, respectively. This results in nested feature subsets without any chance to correct the decision in later steps, causing the performance to be often far from optimal. The idea behind the previous methods aimed at counteracting the nesting effect, can be more efficiently implemented by considering conditional inclusion and exclusion of features controlled by the value of the criterion itself. The Sequential Floating Forward Selection (SFFS) procedure consists of applying after each forward step a number of backward steps as long as the resulting subsets are better than the previously evaluated ones at that level. Consequently, there are no backward steps at all if the performance cannot be improved. The description of the algorithm can be found in (Pudil et al., 1994). Here, we want to compare the performance of the former algorithm with the SFFS algorithm when the evaluation criterion is the Mahalanobis distance d_{ij}: $d_{ij} = (x_i - x_j)S^{-1}(x_i - x_j)$, where S^{-1} is the pooled sample covariance matrix. The Mahalanobis distance incorporates the correlation between features and standardizes each feature to zero mean and unit variance.

3 Data Set

A detailed description of the data set can be found in (Jacobsen et al., 1999). Here we can try to briefly sketch out how the data set was created and what features were used.

The subject of this investigation is the in-service inspection of welds in pipes of austenitic steel. The flaws to be looked for in the austenitic welds are longitudinal cracks due to intergranular stress corrosion cracking starting from the inner side of the tube.

The radio-graphs are digitized with a spatial resolution of 70 mm and a gray level resolution of 16 bits per pixel. Afterwards they are stored and decomposed into various Regions of

Interest (ROI) of 50 x 50 pixel size. The essential information in the ROIs is described by a set of features which are calculated from various image-processing methods.

Images of flaws in welds are radio-graphed by local grey level discontinuities. Subsequently, the morphological edge finding operator, the derivative of Gaussian operator and the Gaussian weighted image moment vector operator are used for feature extraction.

The morphological edge detection operator consists of a combination of morphological operators (i.e. dilation and erosion) which move the gray value edges in an image in different directions. The difference in image *dilation(g(p)) – g(p*) and *g(p) – erosion(g(p))* result in respectively shifted edge-images (where *g(p)* is the original image). After a final minimum operation on both images, the steepest edges remain in the resulting image as the maxima.

The derivative of Gaussian filter is based on a combination of a Gaussian smoothing followed by a partial differentiation of the image in the *x*-and *y*- directions. The result of the filter is chosen as the maximum of the gradient magnitude.

Another filter is designed specially for flaw detection in radio-graphs of welds. This method uses the vector representation of the image and calculates the image moments in an analogous fashion to the model known from mechanics.

A one-dimensional FFT-filter for crack detection problem is also employed. This filter is based on the assumption that the cracks have a preferential horizontal direction in the image. The second assumption is based upon the empirical observation that the half power width of a crack indication is smaller than 300*mm*. The filter consists of a one-dimensional FFT high-pass Bessel operation along a direction orthogonal to that of the cracks, that works with a cutoff frequency of 2 *L/mm*. Normally the half-power width of under-cuts is greater so that this filter suppresses them. This means that it is possible to distinguish between under-cuts and cracks with this FFT-filter. A row-oriented low-pass that is applied to the output of this filter helps to eliminate noise and to point out the cracks more clearly.

Furthermore, a Wavelet filter is also used. The scale representation of the image after the Wavelet transform makes it possible to suppress the noise in the image with a simple threshold operation without losing significant parts of the content of the image. The noise in the image is an interference of film and scanner noise and irregularities caused by the material of the weld.

The features which describe the content of the ROI are extracted from profile plots which run through the ROI perpendicular to the weld. In a single profile plot, the position of a local minimum is detected which is surrounded by two maxima that are as large as possible. This definition varies a little depending on the corresponding image processing routine. A template which is adapted to the current profile of the signal allows us to calculate various features. Additionally, the half-power width and the respective gradients between the local extrema are calculated. To avoid statistical calculation errors, the calculation of the template features is averaged over all of the columns along an ROI. The schematic processing chain is shown in Figure 1. The appearance of the defect in a welding seam and their corresponding profile plots as well as the template used for feature calculation are shown in Figure 2.

The methods outlined here lead to 36 parameters being collected for every ROI. The data set used in this experiment contains features for ROIs from background, crack and undercut regions. The data set consists of altogether 1924 ROIs, 1024 of which are extracted from regions with no disturbance, 465 from regions with cracks and 435 from regions with undercuts.

4 Results

Table 1 illustrates the error rate for the C4.5 classifier when using all features as well as the error rates for different feature subsets calculated based on the CM algorithm. The error rate was estimated using cross-validation. The maximum improvement in accuracy is 1.77 percent for the pruned tree which is obtained as the difference between the error rate when using all features (24.454 in case of the unpruned tree) and the error rate when the feature number is 20 (22.7769 in case of the unpruned tree). Table 3 illustrates the error rate for the C4.5 classifier when using all features as well as error rates for different feature subsets calculated based on the SFFS algorithm. The maximum improvement in accuracy is 2.5 percent for the pruned tree which is obtained as the difference between the error rate when using all features (24.454 in case of the unpruned tree) and the error rate when the feature number is 20 (22.0489 in case of the unpruned tree). To interpret this improvement, we use a classification analysis conducted earlier (Hughes, 1968), where performance actually peaked and then deteriorated as the number of features was increased, see Figure 3 for the CM algorithm and Figure 5 for the SFFS algorithm. We observe similar behavior in our experiments. Classification error is at its minimum in both cases when the feature subset size is 20. This is in contrast to the feature subset size of 30 that C4.5 selects when presented with the entire feature set, with no pre-selection.

It is interesting to note that in the best case both feature subsets have only 12 of 20 features in common, see Figure 4 and Figure 6. This might lead to the slightly better result of the SFFS algorithm. However, both algorithms give the best result for the feature subset with 20 features.

It may be argued that it is not worth doing feature subset selection before tree induction since the improvement in prediction accuracy is not so dramatic. However, the importance of an improvement, however small, clearly depends on the requirements of the application for which the classifier is being trained. We further observed (Figure 4, for 20 features) that about 44% of the total features are used similarly by CM and C4.5, while about 5% of the features are exclusively selected by CM, 41% are exclusively selected by C4.5, and 10% are not used at all. In case of SFFS, we observed (Figure 6, for 20 features) that about 47 % of the total features are used similarly by SFFS and C4.5, while about 5% of the features are exclusively selected by SFFS, 38% are exclusively selected by C4.5, and 10% features are not used at all. Table 2 and Table 4 show that the tree does not necessarily become more compact even if a reduced set of features is used. The tree actually becomes even larger in the case with the best error rate. This observation becomes clear if we look at the mechanism of the tree induction process. The algorithm recursively splits the decision space by axis-parallel decision surfaces during the tree building process. This can result in decision rules of the form e.g. *IF F8≥10 AND .... AND F8≥1 .... THEN Class_X*. One feature is used several times in one path of the decision tree. We therefore cannot draw any useful conclusion about feature set size and its relation to model complexity. We also observe (Figure 4 and Figure 6) that in comparing the two trees generated by C4.5 by one of the feature pre-selection algorithms, the feature used for splitting at the root node changes. This is even true if we compare the first node of the trees generated from the two feature pre-selection algorithm.

5 Conclusion

We have studied the influence of feature subset selection based on two filter approaches to the performance of C4.5. Our experiment was motivated by the fact that C4.5 uses a non-optimal feature search strategy. We used the CM algorithm for feature subset selection which measures importance of a feature based on a contextual merit function. Besides that we used the SFFS feature subset selection algorithm that takes into account feature correlation and has a different search strategy which provides a close to optimal solutions. Our results show that feature subset selection can help to improve the prediction accuracy of the induced classifier. However, it may not lead to more compact trees and the prediction accuracy may not increase dramatically.

The SFFS algorithm performs slightly better than the CM algorithm. Nevertheless, the algorithmic properties of the CM algorithm are better for handling large databases. Both algorithms produce the best result for 20 features.

The main advantage may be that fewer features required for classification can be important for applications such as image interpretation where computational costs for extracting the features may be high and require special purpose hardware. For such domains, feature pre-selection to prune down the feature set size may be a beneficial analysis phase.

Acknowledgement

We thank Pavel Paclik for carrying out the experiment with the SFFS algorithm and Chid Apte for earlier discussions on feature subset selection and for carrying out the experiment with the CM algorithm. For helpful comments on this paper we like to thank Maria Petrou and the anonymous reviewers.

References

Almuallim, H. and Diettrich, T.G. 1994. Learning boolean concepts in the presence of many irrelevant features. Artificial Intelligence, **69**(1-2), 279-305.

Cover, T.M. 1977. On the possible ordering on the measurement selection problem. IEEE Transactions, SMC-7(9), 657-661.

Fukunaga, K. 1990. Introduction to Statistical Pattern Recognition. Academic Press.

Hong, S.J. 1996. Use of contextual information for feature ranking and discretization. IEEE Trans. on Knowledge Discovery and Data Engineering.

Hughes, G.F. 1968. On the mean accuracy of statistical pattern recognizers. IEEE Transactions, IT-14(1), 55-63.

Jacobsen, C., Zscherpel, U., and Perner, P. 1999. A Comparison between Neural Networks and Decision Trees, In, ed. Perner, P. and Petrou, M. Machine Learning and DataMining in Pattern Recognition, Springer Verlag, p. 144-158.

Jain, A. and Zonker, D. 1997. Feature Selection: Evaluation, Application, and Small Sample Performance. IEEE Trans. on Pattern Analysis and Machine Intelligence, **19,** 153-158.

Kira, K. and Rendell, L.A. 1992. The feature selection problem: Traditional methods and new algorithm. In: AAAI-92, Proceedings Nienth National Conference on Artificial Intelligence, AAAI Press/The MIT Press, 129-134.

Kohavi, R. and John, G.H. 1998. The Wrapper Approach. In, ed. Lui, H. and Matoda H. Feature Extraction Construction and Selection, Kluwer Academic Publishers, p. 30-47.

Koller, D. and Sahami, M. 1996. Toward Optimal Feature Selection. In: ed. L. Saitta, Machine Learning, Proceedings of the Thirteenth International Conference (ICML '96),. Morgan Kaufmann, 284-292

Lui, H. and Setiono, R. 1996. A Probabilistic Approach to Feature Selection - A Filter Solution. In: ed. L. Saitta, Machine Learning, Proceedings of the Thirteenth International Conference (ICML '96),. Morgan Kaufmann, 319-327

Lopez de Mantaras, R. 1991. A distance-based attribute selection measure for decision tree induction. Machine Learning, 6, 81-92.

Nadler, M. and Smith, E.P. 1993. Pattern Recognition Engineering, John Wiley&Sons Inc.

Perner, P. 2000. Mining Knowledge in Medical Image Databases, In ed. Belur V. Dasarathy, Data Mining and Knowledge Discovery: Theory, Tools, and Technology, Proceedings of SPIE Vol. 4057, 359-369

Pudil, P, J. Navovicova, and J.Kittler. 1994. Floating search methods in feature selection. *Pattern Recognition Letters,* **15**, 1119-1125.

Quinlan, J.R. 1993. C4.5:Programs for Machine Learning, Morgan Kaufmann

White, A.P, and W.Z Lui. 1994. Bias in the information-based measure in decision tree induction. Machine Learning, **15**,321-329.

Line Number		Crossvalidation		Improvement E=Li-L1
Li i=(1,2,..,6)	Features	unpruned	pruned	pruned
L1	all (C4.5)	24,9610	24,5450	4,1601
L2	10	29,4332	28,7051	-4,1601
L3	15	26,3650	26,4171	-1,8721
L4	20	23,7649	22,7769	1,7681
L5	24	24,4930	23,5049	1,0401
L6	28	25,1170	24,0770	0,4680

Table 1 Error Rate for Different Feature Subsets based on CM Algorithm

Number of Features	10	15	20	24	28	36
Nodes	236	204	178	166	164	161
Edges	237	206	176	137	161	159

Table 2 Number of Nodes and Edges

Line Number		Crossvalidation		Improvement E=Li-L1
Li i=(1,2,..,5)	Features	unpruned	pruned	pruned
L1	all (C4.5)	24,9610	24,5450	2,8600
L2	10	31,8700	27,4050	-2,8600
L3	15	24,9610	24,5970	-0,0520
L4	20	23,2969	22,0489	2,4961
L5	24	25,8450	24,0250	0,5200

Table 3 Error Rate for Different Feature Subsets based on SFFS Algorithm

Number of Features	10	15	20	24		36
Nodes	436	368	366	466		161
Edges	437	370	368	437		159

Table 4 Number of Nodes and Edges

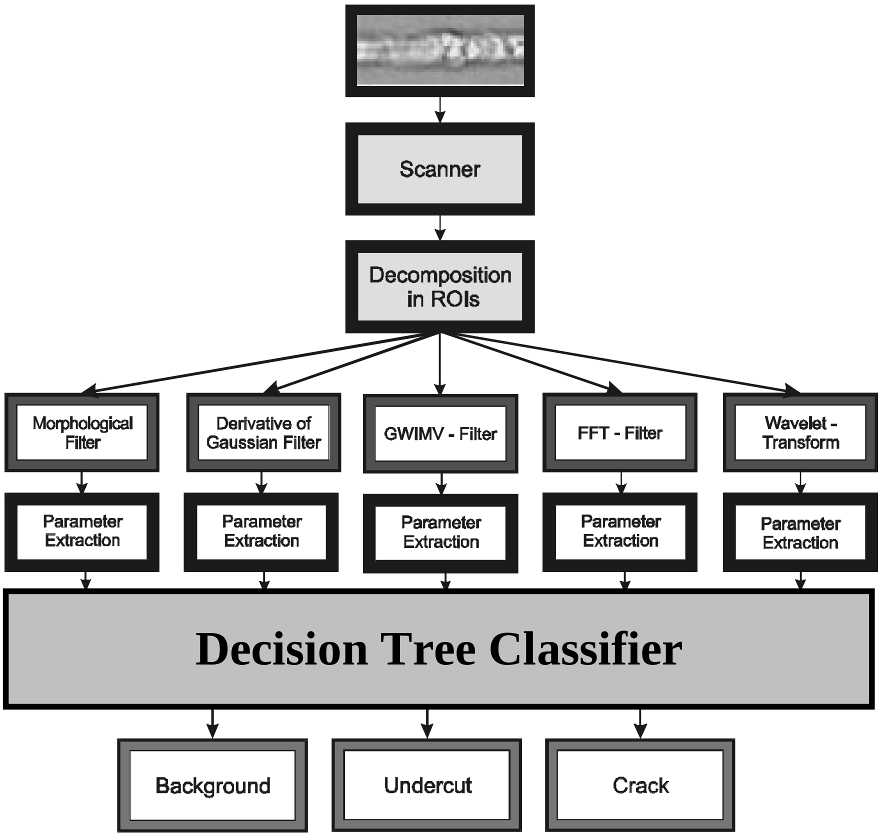


Figure 1 Schematic Representation of Processing Chain

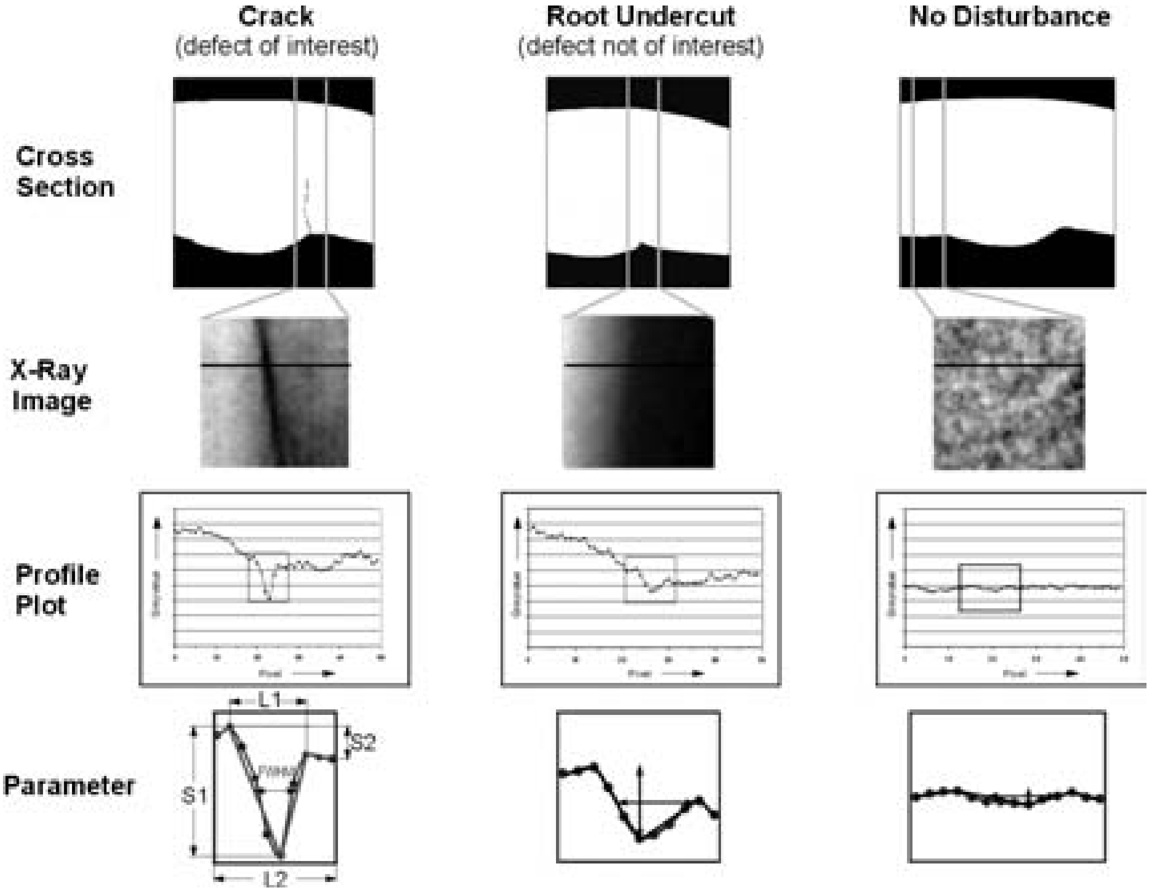


Figure 2 ROI of a crack, an undercut and of a region without disturbance with corresponding cross sections and profile plots

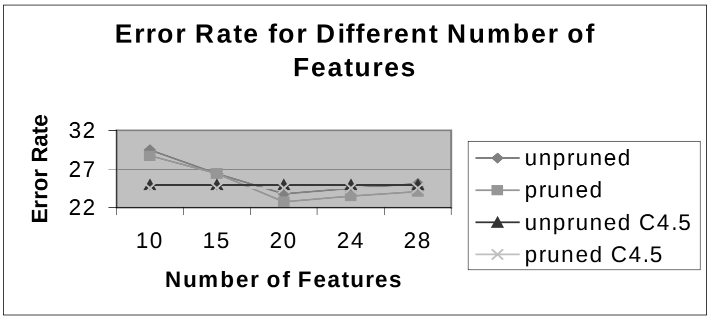
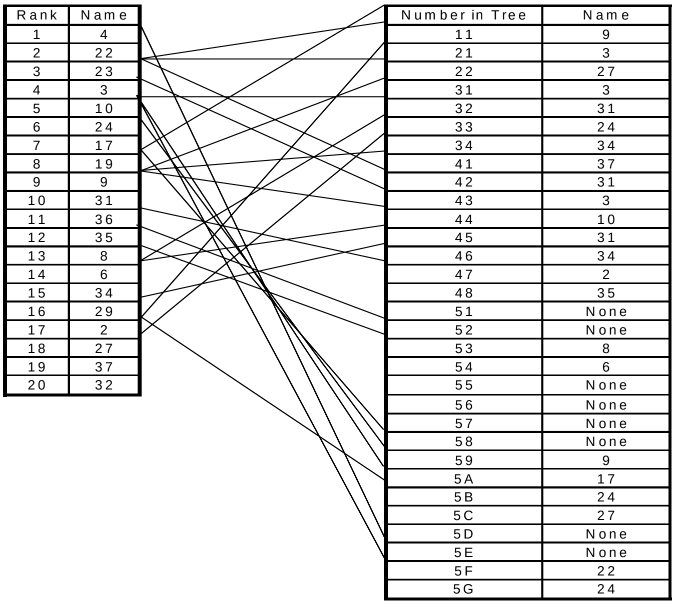


Figure 3 Error Rates for Different Size Feature Sets based on CM Algorithm

Attributes	10	15	20	24	28	C 4.5
2	0	0	1	1	1	1
3	1	1	1	1	1	1
4	1	1	1	1	1	1
5	0	0	0	0	0	1
6	0	1	1	1	1	1
7	0	0		1	1	1
8	0	1	1	1	1	0
9	1	1	1	1	1	1
10	1	1	1	1	1	0
11	0	0	0	0	0	1
12	0	0	0	1	1	1
13	0	0	0	0	0	1
14	0	0	0	0	0	1
15	0	0	0	0	1	1
16	0	0	0	0	0	1
17	0	1	1	1	1	0
18	0	0	0	0	1	1
19	1	1	1	1	1	1
20	0	0	0	1	1	0
21	0	0	0	0	0	1
22	1	1	1	1	1	0
23	0	1	1	1	1	1
24	1	1	1	1	1	1
25	0	0	0	0	0	1
26	0	0	0	0	0	1
27	0	0	1	1	1	1
28	0	0	0	1	1	1
29	0	1	1	1	1	1
30	0	0	0	0	1	1
31	1	1	1	1	1	1
32	0	0	1	1	1	1
33	0	0	0	0	1	1
34	0	0	1	1	1	1
35	1	1	1	1	1	1
36	1	1	1	1	1	1
37	0	0	1	1	1	1
Number used	10	15	20	24	28	31



Rank	Name
1	4
2	22
3	23
4	3
5	10
6	24
7	17
8	19
9	9
10	31
11	36
12	35
13	8
14	6
15	34
16	29
17	2
18	27
19	37
20	32

Number in Tree	Name
11	9
21	3
22	27
31	3
32	31
33	24
34	34
41	37
42	31
43	3
44	10
45	31
46	34
47	2
48	35
51	None
52	None
53	8
54	6
55	None
56	None
57	None
58	None
59	9
5A	17
5B	24
5C	27
5D	None
5E	None
5F	22
5G	24

Figure 4 Ranked Feature and the first 10 Features used by the Decision Tree

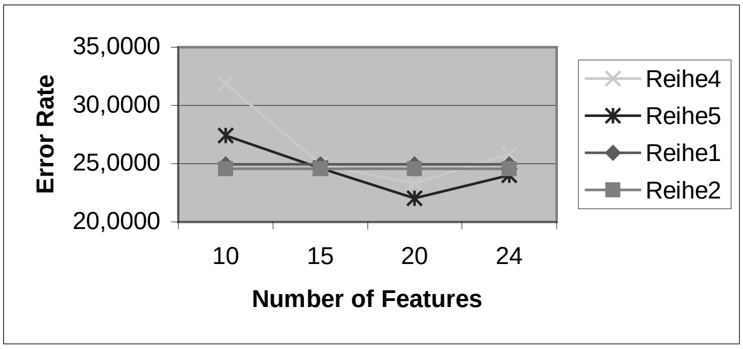
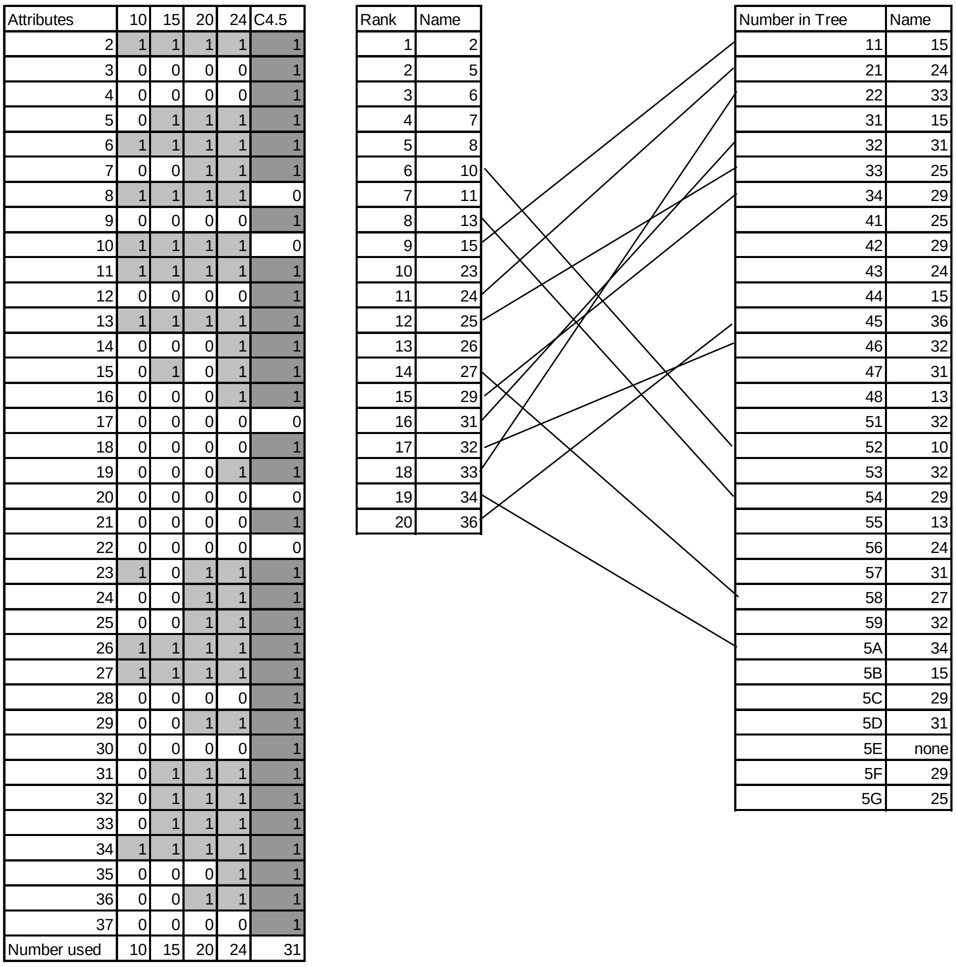


Figure 5 Error Rates for Different Size Feature Subsets based on SFFS Algorithm



Attributes	10	15	20	24	C4.5
2	1	1	1	1	1
3	0	0	0	0	1
4	0	0	0	0	1
5	0	1	1	1	1
6	1	1	1	1	1
7	0	0	1	1	1
8	1	1	1	1	0
9	0	0	0	0	1
10	1	1	1	1	0
11	1	1	1	1	1
12	0	0	0	0	1
13	1	1	1	1	1
14	0	0	0	1	1
15	0	1	0	1	1
16	0	0	0	1	1
17	0	0	0	0	0
18	0	0	0	0	1
19	0	0	0	1	1
20	0	0	0	0	0
21	0	0	0	0	1
22	0	0	0	0	0
23	1	0	1	1	1
24	0	0	1	1	1
25	0	0	1	1	1
26	1	1	1	1	1
27	1	1	1	1	1
28	0	0	0	0	1
29	0	0	1	1	1
30	0	0	0	0	1
31	0	1	1	1	1
32	0	1	1	1	1
33	0	1	1	1	1
34	1	1	1	1	1
35	0	0	0	1	1
36	0	0	1	1	1
37	0	0	0	0	1
Number used	10	15	20	24	31

Rank	Name
1	2
2	5
3	6
4	7
5	8
6	10
7	11
8	13
9	15
10	23
11	24
12	25
13	26
14	27
15	29
16	31
17	32
18	33
19	34
20	36

Number in Tree	Name
11	15
21	24
22	33
31	15
32	31
33	25
34	29
41	25
42	29
43	24
44	15
45	36
46	32
47	31
48	13
51	32
52	10
53	32
54	29
55	13
56	24
57	31
58	27
59	32
5A	34
5B	15
5C	29
5D	31
5E	none
5F	29
5G	25

Figure 6 Ranked Features and the first 10 Features used by the Decision Tree

Figure 1 Schematic Representation of Processing Chain

Figure 2 ROI of a crack, an undercut and of a region without disturbance with corresponding cross sections and profile plots

Figure 3 Error Rates for Different Size Feature Sets based on CM Algorithm

Figure 4 Ranked Feature and the first 10 Features used by the Decision Tree

Figure 5 Error Rates for Different Size Feature Subsets based on SFFS Algorithm

Figure 6 Ranked Features and the first 10 Features used by the Decision Tree