Two-Stage Classification with Automatic Feature Selection for an Industrial Application

Sören Hader[1] and Fred A. Hamprecht[2]

[1] Robert Bosch GmbH, FV/PLF2, Postfach 30 02 40
D-70442 Stuttgart, Germany
[2] Interdisziplinäres Zentrum für Wissenschaftliches Rechnen (IWR),
Universität Heidelberg, D-69120 Heidelberg, Germany

Abstract. We address a current problem in industrial quality control, the detection of defects in a laser welding process. The process is observed by means of a high-speed camera, and the task is complicated by the fact that very high sensitivity is required in spite of a highly dynamic / noisy background and that large amounts of data need to be processed online. In a first stage, individual images are rated and these results are then aggregated in a second stage to come to an overall decision concerning the entire sequence. Classification of individual images is by means of a polynomial classifier, and both its parameters and the optimal subset of features extracted from the images are optimized jointly in the framework of a wrapper optimization. The search for an optimal subset of features is performed using a range of different sequential and parallel search strategies including genetic algorithms.

1 Introduction

Techniques from data mining have gained much importance in industrial applications in recent years. The reasons are increasing requirements of quality, speed and cost minimization and the automation of high-level tasks previously performed by human operators, especially in image processing. Since the data streams acquired by modern sensors grow at least as fast as the processing power of computers, more efficient algorithms are required in spite of Moor's law.

The industrial application introduced here is an automated supervision of a laser welding process. A HDRC (High-Dynamic-Range-CMOS) sensor records a welding process on an injection valve. It acquires over 1000 frames with a resolution of 64×64 pixels per second. The aim is to detect welding processes which are characterized by *sputter*, i.e. the ejection of metal particles from the keyhole, see Fig. 1. These events are rare and occur at most once in a batch of 1000 valves. Potential follow-up costs of a missed detection are high and thus a detection with high sensitivity is imperative, while a specificity below 100% is tolerable.

The online handling and processing of the large amounts of raw data is particularly difficult; an analysis becomes possible if appropriate features are extracted which can represent the process. Of the large set of all conceivable

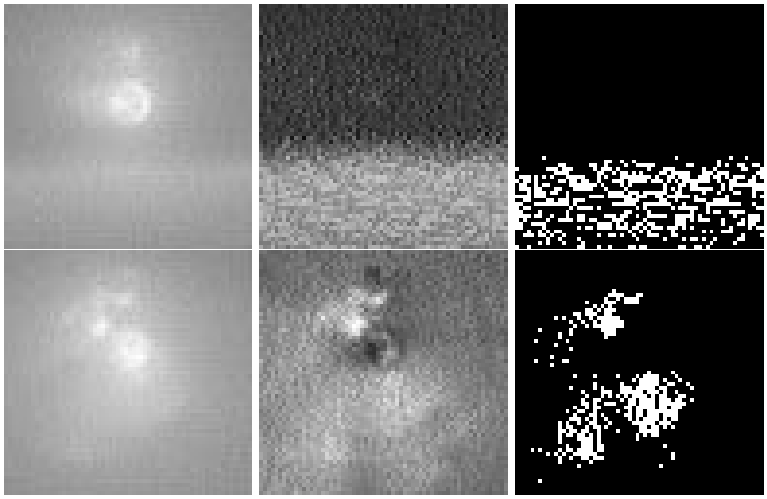

Fig. 1. Top row, left: original frame (64 × 64) from laser welding process, showing a harmless perturbation which should not be detected. Middle: image of the estimated pixel-wise standard deviations, illustrating in which areas the keyhole is most dynamic. Right: pixels which exceed the expected deviation from the mean are marked. Large aggregations of marked pixels are merged to an "object hypothesis". Bottom row: as above, but for original image showing a few sputter that should be detected.

features, we should choose the ones that maximize the classification performance on an entire sequence of images. An exhaustive evaluation of all possible combinations of both features and classifiers is usually too expensive. On the other hand, the recognition performance using a manually chosen feature set is not sufficient in most cases. An intermediate strategy is desired and proposed here: section 2 introduces a two-stage classification system which is optimized using the wrapper approach (section 3) while experimental results are given in section 4.

2 Two-stage classification

2.1 Motivation

While the task is to evaluate the entire sequence of images, we have implemented a divide-and-conquer strategy which focuses on individual images first. In particular, we use a very conservative classifier on individual images: even if there is only a weak indication of an abnormality, the presumed sputter is segmented from the background and stored as an *object hypothesis*. Evidence for a sputter is substantiated only if several such hypotheses appear in consecutive frames.

The advantage of a simple classification in the first stage is the fast evaluation and adaptation of the classifier. The second stage aggregates classifica-

tions derived from individual images into an overall decision with increased reliability.

2.2 First stage – object classification

In the first stage, object hypotheses from single images are extracted and classified.

In particular, an image of pixel-wise means and an image of pixel-wise standard deviations are computed from the entire sequence. Deviations from the mean, which are larger than a constant (e.g. $\in [2.0, 4.0]$) times the standard deviation at that pixel are marked as suspicious (Brocke (2002), Hader (2003)). Sufficiently large agglomerations of suspicious pixels then become an object hypothesis $O_{t,i}$ with indices for time t and object number i. Next, features such as area, eccentricity, intensity, etc. (Teague (1980)) are computed for all object hypotheses.[1] Based on these features, we compute (see section 2.4) an index $d(O_{t,i}) \in [0, 1]$ for membership of object hypothesis $O_{t,i}$ in class "sputter".

2.3 Second stage – image sequence classification

The first stage leaves us with a number of object hypotheses and their class membership indices. Sputters appear in more than one consecutive frame, whereas random fluctuations have less temporal correlation. The second stage exploits this temporal information by aggregating the membership indices into a single decision for the entire sequence as follows: for each frame, we retain only the highest membership index: $d_t := \max_i d(O_{t,i})$. If there is no hypothesis in a frame, the value is set to 0. The d_t can be aggregated using a variety of functions. We use a sliding window located at time t, and apply the $\sum, \prod, \min$ operators to the indices $d_t, \ldots, d_{t+w-1}$ to obtain aggregating functions $a_w(t)$. The length of the time window w is arbitrary, but should be no longer than the shortest sputter event in the training database. The largest value of the aggregate function then gives the decision index for the entire sequence,

$$d_{\text{sequence}} = \max_{t \in T} a_w(t) \tag{1}$$

If d_{sequence} exceeds a threshold Θ, the entire sequence is classified as defective, otherwise as faultless. The optimum value for the threshold Θ depends on the loss function, see section 3.

[1] This list of features is arbitrarily expandable and previous knowledge on which (subset of) features are useful is not necessary, see section 3.1.

2.4 Polynomial classifier

The choice of the classifier used in the first stage is arbitrary. We use the polynomial classifier (PC, Schürmann (1996)) which offers a high degree of flexibility if sufficiently high degrees are used. Since it performs a least-squares minimization, the optimization problem is convex and its solution unique. Training is by solving linear system of equations and is faster than that of classifiers like multilayer perceptrons or support vector machines (LeCun et al. (1995)), which is important in case the training is performed repeatedly such as a wrapper optimization (section 3.1). PCs have essentially only one free parameter, the polynomial degree.

In the development stage, a tedious manual labeling of image sequences is required to assemble a training set. Based on an initial training set and the resultant classifier, further sequences can be investigated. The variance of predictions for single object hypotheses can be estimated and those for which a large variance is found can be assumed to be different from the ones already in the training set and added to it. In particular, under a number of assumptions (uncorrelated residuals with zero mean and variance σ^2) the variance of a prediction can be estimated by $\sigma^2 x^T (X^T X)^{-1} x$ where X is the matrix of all explanatory variables (features and monomials formed from these) for all observations in the training set, and x is the new observation (Seeber and Lee (2003)).

3 System optimization

As stated above, sensitivity is of utmost importance in our application, while an imperfect specificity can be afforded. These requirements are met by optimizing the detection threshold Θ such that the overall cost is minimized. The losses incurred by missed detections or false positives are given by $L_{NIO,IO}$ and $L_{IO,NIO}$, respectively, with the former much larger than the latter.

It is customary to arrange the loss function in a matrix as shown below:

$$L = \begin{vmatrix} L_{IO,IO} & L_{IO,NIO} \\ L_{NIO,IO} & L_{NIO,NIO} \end{vmatrix}, \; L_{IO,IO} = L_{NIO,NIO} = 0, \; L_{IO,NIO} \ll L_{NIO,IO}$$

The first index gives the true class, the second one the estimated class, with IO faultless, and NIO defective. The aim is to find a decision function which minimizes the Bayes risk $r = \mathbb{E}\{L\}$. A missed NIO part makes for a large contribution to the risk $\hat{r}$.

The generalization error of a given feature subset and classifier is estimated from the bins that are held out in a k-fold cross-validation (CV). 5- or 10-fold CV is computationally faster than leave-one-out and is a viable choice in the framework of a wrapper algorithm; moreover, these have performed well in a study by Breiman and Spector (1992).

3.1 Wrapper approach

We see great potential in the testing of different feature subsets. In earlier applications the filter approach (which eliminates highly correlated variables or selects those that correlate with the response) was the first step in finding the relevant features. The filter approach attempts to assess the importance of features from the data alone. In contrast, the *wrapper approach* selects features using the induction algorithm as a black box without knowledge of feature context (Kohavi and John (1997)). The evaluation of a large number of different subsets of features with a classifier is possible only with computationally efficient procedures such as the PC. We use the wrapper approach to simultaneously choose the feature subset, the polynomial degree G, the operator in the aggregation function a, the window width w and the threshold Θ. Evaluating a range of polynomial degrees $1, \ldots, G$ is expensive; in section 3.3 we show how PCs with degree $< G$ can be evaluated at little extra cost.

3.2 Search strategies in feature subsets

The evaluation of all 2^n combinations of n individual features is usually prohibitive. We need smart strategies to get as close as possible to the global optimum without an exhaustive search. Greedy sequential search strategies are among the simplest methods, with two principal approaches, sequential forward selection (SFS) and sequential backward elimination (SBE). SFS starts with an empty set and iteratively selects from the remaining features the one which leads to the greatest increase in performance. Conversely, SBE begins with the complete feature set and iteratively eliminates the feature that leads to the greatest improvement or smallest loss in performance. Both SFS and SBE have a reduced complexity of $\mathcal{O}(n^2)$. Both heuristics can miss the global optimum because once a feature is selected/eliminated, it is never replaced again.

A less greedy strategy is required to reach the global optimum. In particular, locally suboptimal steps can increase the search range. We use a modified BEAM algorithm (Aha and Bankert (1995)) in which not only the best, but the q best local steps are stored in a queue and explored systematically. Deviating from the original BEAM algorithm, we allow either the adding of an unused feature or the exchange of a selected with an unused feature.

Another global optimization method are genetic algorithms (GAs), which represent each feature subset as member of a population. Individuals can mutate (add or lose a feature) and mate with others (partly copy each other's feature subsets), where the probability of mating increases with the predictive performance of the individuals / subsets involved. It is thus possible to find solutions beyond the paths of a greedy sequential search. A disadvantage is the large number of parameters that need to be adjusted and the suboptimal performance that can result if the choice is poor.

3.3 Efficiency

The analysis of the runtime is important to understand the potential of the PC for speed-up. A naive measure of the computational effort is the total count of multiplications. Although it is just a "quick and dirty" method ignoring memory traffic and other overheads, it provides good predictions.

Solution of the normal equation

$$\mathbb{E}\{xx^T\} \cdot A = \mathbb{E}\{xy^T\} \tag{2}$$

with x a column vector specifying the basis functions (i.e. the monomials built from the original features) of an individual observation, y a vector which is $[1\ 0]^T$ for one class and $[0\ 1]^T$ for the other, and A the coefficient matrix. The expectation values are also called moment matrices.

The computational effort mainly consists of two steps: estimation of the moment matrix $\mathbb{E}\{xx^T\}$ and its inversion. The former requires D^2N multiplications, with N the number of observations and $D = \binom{F+G}{G}$ the dimension of x, that is the feature space obtained by using all F original features as well as all monomials thereof up to degree G.

In CV, the data is partitioned into k bins; accordingly, the $N \times D$ design matrix X can be partitioned into $N_i \times D$ matrices X_i, with $\sum_{i=1}^{k} N_i = N$. The moment matrices are estimated for each bin separately by $X_i^T X_i$. For the jth training in the course of a k-fold CV, the required correlation matrix is obtained from

$$X_{-j}^T X_{-j} = \sum_{i \neq j} X_i^T X_i \tag{3}$$

that is, $D \times D$ matrices are added only.

In summary, while the correlation matrices need to be inverted in each of the k runs in a k-fold CV (requiring a total of $k\frac{2}{3}D^3$ multiplications for a Gauss-Jordan elimination), they are recomputed at the cost of a few additions or subtractions only once the correlation matrices for individual bins have been built (requiring a total of D^2N multiplications).

In addition, once the correlation matrix for a full feature set F and polynomial degree G has been estimated, all moment matrices for $F' \subseteq F$ and $G' \leq G$ are obtained by a mere elimination of appropriate rows and columns.

4 Experimental results

The system has been tested on a dataset of 633 IO and 150 NIO image sequences which comprise a total of 5294 object hypotheses that have been labeled by a human expert. A large part of the IO sequences selected for training were "difficult" cases with sputter look-alikes. The loss function used was $L_{IO,NIO} = 1$ and $L_{NIO,IO} = 100$ and generalization performance was estimated using a single 10-fold CV. A total of 19 features were computed for

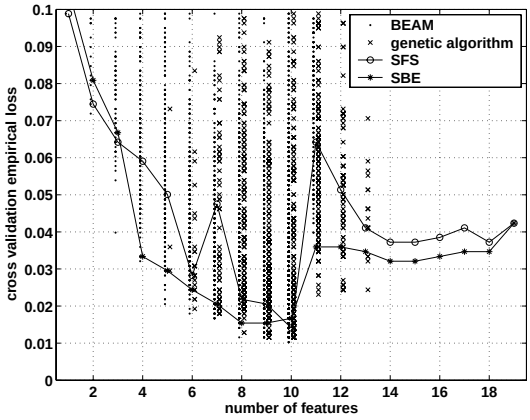

Fig. 2. Each point gives the generalization performance, as estimated by CV, for a particular subset of features and an optimized classifier. For a given subset, all classifier parameters such as aggregation function operator and its window width, degree of polynomial, and threshold Θ, were optimized using a grid search.

each object hypothesis. The four subset selection strategies described were tested. For the modified BEAM algorithm, the parameter $q = 5$ and 20 generations were used. The GA ran for 50 generations with 60 individuals each.

Results are shown in Fig. 2. SBE works better than SFS on average, though their best results are similar ($\hat{r} = 0.015$ and 0.014). BEAM and GA offer minor improvements ($\hat{r} = 0.010$ and 0.012) only.

Surprisingly, the final optimized system recognizes individual object hypotheses with a low accuracy: $\hat{r} = 0.341$ with $L_{Non-Sputter,Sputter} = 10$ and $L_{Sputter,Non-Sputter} = 1$. The high performance obtained in the end is entirely due to the temporal aggregation of evidence from individual frames.

Figure 3 shows the results obtained when the membership index $d(O_{t,i})$ is not given by the object estimate obtained from the PC, but by the lower bound of an interval estimate to reflect the strongly asymmetric loss function. Overall classification accuracy is not improved, but the magnitude of the interval can help identifying sequences that ought to be labeled manually and should be included in future training sets.

5 Conclusion and outlook

Since the number of objects, N, is typically much larger than the number of basis functions, D, the most expensive part in training a PC is the computation of the correlation matrix and not its inversion. Recomputations of the former can be avoided in the framework of cross-validation, as illustrated in section 3.3 For our particular data set, advanced subset selection strategies did not lead to a much improved performance.

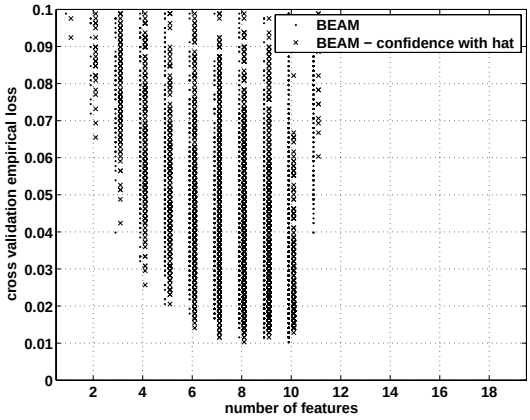

Fig. 3. Results obtained when replacing object estimates of class membership in individual images with the lower bound of interval estimates, see section 2.4.

Even though all features computed on object hypotheses were chosen with the aim of describing the phenomenon well, the generalization performance varies greatly with the particular subset that is chosen in a specific classifier. A systematic search for the optimal subset is thus well worth while, and is made possible by the low computational cost of the PC which allows for a systematic joint optimization of parameters and feature subset.

References

AHA, D.W. and BANKERT, R.L. (1995): A comparative evaluation of sequential feature selection algorithms. In: D. Fischer and H. Lenz (Eds.): *Fifth International Workshop on Artificial Intelligence and Statistics*. 1–7.

BREIMAN, L. and SPECTOR, P. (1992): Submodel selection and evaluation in regression: The X-random case. *International Statistical Review, 60, 291–319*.

BROCKE, M. (2002): Statistical Image Sequence Processing for Temporal Change Detection. In: L. Van Gool (Eds.): *DAGM 2002, Pattern Recognition*. Springer, Zurich, 215–223.

HADER, S. (2003): System Concept for Image Sequence Classification in Laser Welding. In: B. Michaelis and G. Krell (Eds.): *DAGM 2003, Pattern Recognition*. Springer, Magdeburg, 212–219.

KOHAVI, R. and JOHN, G.H. (1997): Wrappers for Feature Subset Selection. *Artificial Intelligence, 97(1-2), 273–324*.

LECUN, Y. et al. (1995): Comparison of learning algorithms for handwritten digit recognition. In: F. Fogelman and P. Gallinari (Eds.): *International Conference on Artificial Neural Networks*. 53–60.

SCHÜRMANN, J. (1996): *Pattern Classification*. John Wiley and Sons, Inc., New York.

SEEBER, G. and Lee, A. (2003): *Linear Regression Analysis*. Wiley-Interscience.

TEAGUE, M.R. (1980): Image Analysis via the General Theory of Moments. *Opt. Soc. of America, 70, 920–930*.