# A TOOLBOX FOR ANALYSIS AND VISUALIZATION OF SENSOR DATA IN SUPERVISION

T. W. Rauber, M. M. Barata and A. S. Steiger-Garção
Universidade Nova de Lisboa – Faculdade de Ciências e Tecnologia
Departamento de Informática – Intelligent Robotics Group
2825 Monte de Caparica – Portugal
Tel. +351-1-2953220 – Fax +351-1-2955641 – E-Mail tr@fct.unl.pt

**Abstract.**

Fault diagnosis based on sensors is a data analysis problem. From the huge amount of incoming data of the sensors, only a small fraction may be helpful for the characterization of a certain state of a production process. The researcher often faces the problem that he cannot use the data in its totality. He rather wants to know if a sensor can be used for diagnosis and which specific sensor data is best suited for characterizing a class of situations.

As a premise for the following work we assume that situations of a process are characterized by multidimensional continuous feature vectors. The task is now to analyze these process states in order to be able to make predictions or statements about the actual states of a process.

The data may be of quite different nature, like visual data, sound patterns or transduced sensor data. The analysis tool should therefor be flexible in respect to that variety of different data and make as few assumptions as possible about their statistical behavior.

In this paper a set of tools is presented which focuses mainly on the following tasks:

- Feature selection:
  The dimension of the vector of the samples that characterize a process state is reduced to a reasonable size. Several algorithms are proposed.
- Error estimation:
  With the reduced feature set, the performance of a classifier based on these features is forecasted. This means that we make predictions on how well a classification of unknown process situations can be done in the future.
- 2-D Visualization:
  The distribution of the classes in the feature space can be visualized in the x-y-plane with the help of a non-linear mapping technique. Also the behavior of the estimated error in respect to the number of selected features can be plotted. Furthermore linear correlation between 2 different features can be tested and visualized.

The main philosophy of this analysis tool is to give the researcher an idea about his sensor data: Is it good for characterizing a certain state? Which particular subset of the sensor data is better than the other? How close are certain features related to others? Etc.

Several case studies in a test bed of semiprofessional NC-machines will illustrate the usefulness of the toolbox.

**Key Words.**

Supervision, monitoring, statistical pattern recognition, feature selection, error estimation, supervised learning, visualization.

# 1 Introduction

Researchers in the area of process supervision often face the problem of unknown parameters for good machine behavior. How can faults be detected if their description is missing or how a fault is generally defined? The information of the actual state of a production process is normally acquired by sensors. And the nature of the sensor data is normally non-symbolic, numerical data in the form of multidimensional feature vectors. An essential task is the analysis of the vast amount of incoming sensor data in order to be able to extract structural knowledge. A data reduction must be performed by feature selection techniques. Which characteristic parts of the sensor data are best suited to describe a certain behavior of a machine? Then, how can these characteristic parts be used to identify a particular situation of a process? It should also be possible to investigate the topological structure of the sensor data with the help of visualization programs.

This leads to a pattern recognition problem. A supervised learning is performed. A particular production phase is sensed and the sensor data is labeled with that phase. Classes are defined as machine states and samples as snapshots of the machine state while it is working. It is now necessary to analyze that labeled data.

As an example take the sound a NC machine produces during its operation. A microphone receives the sound, an A/D converter generates numerical information and a Fourier transform changes this information to the frequency domain. If the signal is sampled at only a moderate rate one typically receives buffer sizes of e.g. 256 real numbers. High amounts of noise can be expected to be hidden in the feature vector and only a few values may be valuable for classification tasks.

But not only particular information about class membership of a sensed value can be inferred from a classifier. It could furthermore be concluded that a particular sensor is not suited to characterize particular machine states. In this case the statistical analysis will not be able to find discriminating features for the classes that were a priori defined by the machine operator. In either the case, that classes can be well distinguished or are overlapping, the data analysis permits helpful decision support.

The first step of the sensor data reduction is feature selection. Since the supervision of machines is a generic problem, the feature selection algorithm should be of general purpose nature. From the initial dimension of the feature vectors (e.g. 256) only a small amount will be used for classification purposes (e.g. 10).

With the reduced feature set a classifier is designed, e.g a Bayesian model, non-parametric model or connectionist (neural network) model. One wishes to know how well the chosen feature set will perform for classification. This is the task of the error estimation module.

After the reduced feature set proves to be reliable for classification, a further data reduction must be performed. From all samples that were provided, a representative part will be retained for future classification task. For the Bayesian model this step would mean the parameter learning, for the neural network a weight learning. In the case of non-parametric models a prototype representation is useful. The whole sample set is transformed into a much smaller set of representative vectors. We propose the work of Kohonen for solving the learning problem of the samples [1]. In the supervised case his *Learning Vector Quantizer (LVQ)* is able to solve the learning problem by condensing the training data to a small set of codebook vectors.

Another important part of our work is the visualization of the samples composed of feature that were taken during the learning and identification. Since these feature vectors are normally of a dimensionality which even in a selected form are higher than the human imaginative capacity, a mapping to 2 dimensions must be performed. This allows now to get an idea of how well certain features can distinguish the classes of the process states. Sammon's algorithm [4] maps a set of multidimensional class distributions to the x-y-plane.

All these conceptual ideas lead to a toolbox in form of a software package for the analysis and visualization of the sensor data. Public domain software as well as own modules are merged into a tool which should support people working in the area of machine monitoring and supervision. A free access to the module will be provided to everyone who wishes to experiment its functionality. It should be understood as a program in steady evolution.

# 2 Data Acquisition

The first step in sensor data analysis is the definition and acquisition of the samples which describe the process situation. We will use the sensor data that is acquired during a production process of a NC machine. One sensor could be the Fourier transform of the sound that the spindle rotation of a NC machine produces during a production process. At this example the general philosophy of the work is illustrated. The task is to transduce a particular state of the machine to a sound pattern that consists of 256 continuous features. Classes are defined as different working states. A class could be e.g. "drilling of a hole into steel at 300 rpm in normal operation" or "drilling of a hole into steel at 300 rpm with tool defect". The acquisition process generates the multidimensional feature vector and labels it with the respective class (normal or defect).

Other sensors are the vibrations sensed along the axes directions and the mains current consumption of the machine tool.

The interface to the analysis module could be on-line or via files. The files have the advantage that they can keep the input information for later experimentation with different strategies.

# 3 Feature Selection

A horizontal data reduction has to be performed, i.e. from the 256 features only a reduced number will be retained; those features which posses the highest discriminative power will be used later for the classification of the process states. A feature selection should be able to find the features which contain the most significant information of the original data. The general formulation of the feature selection problem is therefor:

For the original data vector $\underline{x} = (x_1, \ldots, x_D)$ find one $\underline{x}' = (x_1, \ldots, x_d)$ with a reduced dimension $d < D$ which represents the classes more compactly. See e.g. [5].

One of the fundamental problems of feature selection is that the optimal solution cannot be found by exhaustive search. Already for a relatively small dimension, the number of possible combinations is impossible to analyze. Therefor in most cases approximative heuristics must be implemented. Also the underlying statistical distribution of the features is not known a priori. This means that furthermore assumptions must be made, if parametric models are used. Feature selection consists of three components: a.) search strategy, b.) selection criterion and c.) stopping rule.

We propose three different strategies for feature selection, each with its necessary assumptions and properties. Each strategy has its own separability measurement or criterion function which should give a high value if the classes separate well and a low value if they overlap.

## 3.1 Strategy I: Univar

Since the particular features are not independent among each other the analysis must normally be performed in a multivariate context. The first strategy that we propose ignores that interdependency of the features and analyzes only one at a time. A separability criterion for the classes is calculated and the features are ranked following this criterion. The class separability measure was proposed in [6].

### 3.1.1 Assumptions

i.) Features are independent.
The class-conditional probabilities for a given class $\omega$ of two or more features are the product of the class-conditional probabilities of the single features:

$$p((x_1, \ldots, x_d) | \omega) = \prod_{k=1}^{d} p(x_k | \omega)$$

ii.) Each feature $x_k$ is distributed in a unimodal way with the mean $\mu_k$ and standard deviation $\sigma_k$. This two parameters are estimated from the sample mean and variance respectively.

### 3.1.2 Criterion function

For a class $\omega_i$ and a feature value $x$ the *normalized probabilistic distance* between the class mean and the feature value is defined as:

$$d_i(x) = \frac{|x - \mu_i|}{\sigma_i}$$

For $d_i(x) = d_j(x) = d$ one can verify that the following holds if $x$ is supposed to be located between $\mu_i$ and $\mu_j$ supposing that $\mu_i < \mu_j$:

$$d = \frac{\mu_i - \mu_j}{\sigma_i + \sigma_j}$$

For decreasing $\sigma_i, \sigma_j$ and a growing distance between $\mu_i$ and $\mu_j$ we notice that $d$ is growing. Logically big values of $d$ express a great interclass distance between the two classes $\omega_i$ and $\omega_j$. They are a thumbnail guess for "good" or "bad" features.
If Chebychev's inequality

$$P\{|x - \mu_i| \geq \varepsilon\} \leq \frac{\sigma_i^2}{\varepsilon^2}$$

is applied to the normalized distance $d_i(x)$ and for that $x$ where the distances $d_i(x)$ and $d_j(x)$ are equal, we obtain

$$P\{d_i \geq d\} \leq \frac{1}{d^2} \quad .$$

This probability should be as small as possible. A small probability is equivalent to the fact that a value of the feature lies nearer to its own class than to the other class.
Hence the following term should also be minimal for a highly discriminative feature:

$$\frac{1}{d^2}$$

This allows to establish a quantitative power of class separability measure $\Delta$ between two classes:

$$\Delta(\omega_i, \omega_j) = 1 - \left(\frac{\sigma_i + \sigma_j}{\mu_i - \mu_j}\right)^2 \quad .$$

This distance function expresses how well a certain feature can distinguish two classes. The value 1.0 is the optimum and says that the two classes are well separated. (Negative values can occur for very high class overlapping).
Calculating the average of all mutual distances between all classes we finally obtain the feature selection criterion for the *Univar* algorithm:

$$Criterion(Univar) = \frac{2}{c(c-1)} \sum_{i=1}^{c-1} \sum_{j=i+1}^{c} \Delta(\omega_i, \omega_j)$$

where $c$ is the number of classes. Thus it is possible to rank the features following this quality criterion.
Figure 1 illustrates the calculus of the separability criterion for the example of an unimodal Gaussian distribution.

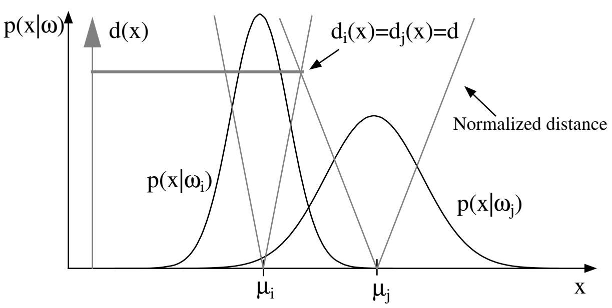


Figure 1: Normalized probability distances between two classes.

### 3.1.3 Properties

The biggest advantage of this feature selection criterion is the speed of calculation. The computational costs are very low, because features can sequentially be considered and ranked. As a disadvantage one can state that highly correlated features are not detected. The best and the second best features could be linearly correlated, but would both be classified as highly discriminant by the algorithm. Another disadvantage is that single features may be ranked with a low discriminative power, although the multivariate combinations of the single features might prove to be very discriminative.

### 3.1.4 The Algorithm

```
1.)For all features:
     Calculate criterion(Univar).
2.)Order all features following the crite-
    rion calculated in 1.
3.)Choose the best d features.
```

The number $d$ could be specified by the user a priori, or it could be determined in function of the error estimation, e.g. choose that number which gives the lowest estimated error for the classification task, see section 4.

## 3.2 Strategy II: M-Covar

This kind of feature selection strategy takes into consideration the dependency among the features. A feature vector is considered as a multivariate random variable. We expect that some features contain much information about the class distributions. On the other hand there are features that do not contribute much information if we want to separate the classes.

The task of *multiple covariance analysis* is to make statements about additional features, called *predictors* that are compared to already selected features, called *controls*. The theory of this strategy cannot be explained in much detail. Statistical background is used from multivariate analysis of variance (MANOVA), multivariate normal distributions and multiple regression analysis. For details consult e.g. [7]. We will just point out how the feature selection criterion is calculated for each feature.

In general we consider the pool of already selected features (controls) and then calculate for each of the features that are not yet selected the *univariate F-Ratio*. Thus we have *d-1* control variables and *1* predictor variable.

### 3.2.1 Assumptions

i.) Features are multivariate normally distributed.

### 3.2.2 Criterion function

Let the following variables be valid:

$c$: Number of classes $k = 1 \ldots c$
$N$: Number of all samples
$N_k$: Number of samples of class $k$
$p$: Number of control variables (= number of already

selected features)

$q$: Number of all predictor variables (always = 1)

$\underline{x}_{k_i} = (x_{1,k_i}, \ldots, x_{p,k_i}, x_{q,k_i})$ : $k_i$-sample of class $k$ which consists of the $p$ already selected features and the control feature $q$.

$\underline{\mu}_k$: local centroid for class $k$

$\underline{\mu}$: global centroid for all classes

$$\underline{\mu}_k = \frac{1}{N_k}\sum_{i=1}^{N_k} \underline{x}_{k_i} \qquad \underline{\mu} = \frac{1}{N}\sum_{k=1}^{c}\sum_{i=1}^{N_k} \underline{x}_{k_i}$$

Let the "total" matrix $T$ be composed of the sum of "among-classes" matrix $A$ and the "within-classes" matrix $W$ satisfying the equation $T = A + W$.

$$T = \sum_{k=1}^{c}\sum_{i=1}^{N_k} \underline{x}_{k_i}\underline{x}_{k_i}^T$$

$$A = \sum_{k=1}^{c}\sum_{i=1}^{N_k} (\underline{\mu}_k - \underline{\mu})\,(\underline{\mu}_k - \underline{\mu})^T$$

$$W = \sum_{k=1}^{c}\sum_{i=1}^{N_k} (\underline{x}_{k_i} - \underline{\mu}_k - \underline{\mu})\,(\underline{x}_{k_i} - \underline{\mu}_k - \underline{\mu})^T.$$

Split the matrix $T$ and the matrix $W$ following the schema

$$T = \left[\begin{array}{c|c} [T_{11}] & [T_{12}] \\ \hline [T_{21}] & [T_{22}] \end{array}\right] \qquad W = \left[\begin{array}{c|c} [W_{11}] & [W_{12}] \\ \hline [W_{21}] & [W_{22}] \end{array}\right]$$

where $(p-1, p-1)$, $(p-1, 1)$, $(1, p-1)$, $(1, 1)$ are the dimensions of the sub-matrices $T_{11}, T_{12}, T_{21}, T_{22}$ respectively (analogously for $W$).

The *total residual matrix* $T_{2\cdot 1}$ and the *within residual matrix* $W_{2\cdot 1}$ are defined as:

$$T_{2\cdot 1} = T_{22} - T_{21}T_{11}^{-1}T_{12} \qquad W_{2\cdot 1} = W_{22} - W_{21}W_{11}^{-1}W_{12}$$

Now we can define the $j$-th *univariate F-Ratio* as

$$F_j = \frac{(t_{2\cdot 1_{jj}} - w_{2\cdot 1_{jj}})/f_1}{(w_{2\cdot 1_{jj}}/f_2)}$$

with $f_1 = c - 1$ being the degrees of freedom for the nominator and $f_2 = N - c - p$ being the degrees of freedom for the denominator.

For the number of predictors $q = 1$ the index $j$ can only have the value *1* as well.

Finally we define the feature selection criterion for the M-Covar algorithm as:

$$Criterion\,(M\text{-}Covar) = F_1$$

### 3.2.3 Properties

The computational costs of the calculus is relatively high compared to the anterior feature selection algorithm. Especially the matrix calculus for $A$, $W$, the inversion of the sub-matrices $T_{11}$, $W_{11}$ and their multiplication with the other sub-matrices are expensive operations. The main benefit of the algorithm is the multivariate character that it has. Highly correlated features that show a high discriminative power for the *Univar* algorithm are represented by only one discriminative feature. A feature that is a predictor and that has a control to which it is highly correlated will have a low univariate F-Ratio.

### 3.2.4 The Algorithm

```
1.)  Choose one arbitrary feature as the
     first control variable
2.)  Repeat
     2.1)For all unselected features u:
       2.1.1)Choose u as predictor variable
       2.1.2)Calculate criterion(M-Covar)
       2.1.3)Order all features following
          the criterion calculated in 2.1.2
     2.2)Join the feature with the highest
       value for the criterion to the con-
       trols and to the selected features.
       (If the second feature has been
       selected, substitute the first one by
       the second one; the first was arbi-
       trary)
     Until (the desired number of selected
       features is reached) or (the maximum
       of all calculated criteria falls short
       of a threshold (typically 4.0))
```

## 3.3 Strategy III: MinErr

Similar to the anterior method we keep a pool of already selected features, and the rest of the features are possible candidates. The philosophy of this strategy is to join that feature to the selected ones that gives the lowest estimated error of classification. The error estimation is performed with the already selected features together with the candidate. Each unselected feature is considered as candidate. The error estimation is independent of the selection strategy. We will use the *leave-one-out* method in a non-parametric model. The error estimation is presented in the next section.

### 3.3.1 Assumptions

No assumptions have to be made about the classes and the features that define them.

### 3.3.2 Criterion function

Design a classifier from the samples $\underline{x}_1, \ldots \underline{x}_N$ using a subset of $j$ features from all features:.

$$\underline{x}_i = (x_{is_1}, \ldots, x_{is_j})$$

Estimate the error $\hat{e}(j)$ that the classifier will exhibit during its functioning using a standard error estimation technique. The selection criterion is then:

$$Criterion\,(MinErr) = \hat{e}(j)$$

### 3.3.3 Properties

This method is multivariate in the sense that it estimates the error of the classifier taking into consideration the whole set of the already selected features plus the candidate for which the case is considered at the moment. The computational costs depend exclusively on the error estimation procedure. If we use e.g. the *leave-one-out* technique and iteratively select the next feature, the costs could become somewhat expensive.

#### 3.3.4 The Algorithm

```
1.)  Repeat
     1.1)For all unselected features u:
       1.1.1)Choose u as candidate
       1.1.2)Estimate error of a classifier
          that uses the already selected fea-
          tures together with the candidate
       1.1.3)Order all features following
          the criterion calculated in 1.1.2
     1.2)Join the feature with the lowest
       error for the criterion to the
       selected features.
     Until (the desired number of selected
       features is reached)
```

# 4 Error Estimation

*Hold-out* error estimation divides a sample set into a training set and a test set. A classifier is induced from the training set and the test set is then fed to the classifier. The error rate of the test is reported. That splitting into training and test set, the classifier generation and the test can be repeated several times. Usually the overall estimated error is then the average of all these splitting passes.

The *leave-one-out* error estimation is given when we repeat this splitting and testing as many times as the number of samples. We pick out one single sample, design a classifier with the rest of the samples and test if the classifier identifies correctly or not. This process is repeated for all samples and the absolute number of occurred misclassifications is divided by the number of samples. The following schema lists the leave-out-our error estimation technique:

```
Let n be the number of samples of the data
    set S.
Let e be the number of errors.

Algorithm: Leave-one-out(S)

1.) e ← 0
2.) For all samples i of the data set i=1…n
      1.1) Generate classifier with all sam-
        ples j of the data set j=1…n, j≠i;
      1.2) Classify sample i
      1.3) If i was misclassified increment
        e: e ← e+1
3.) Estimated error rate  ← e/n.
```

The leave-one-out error estimation is slightly pessimistically biased. The computational burden is relatively high, it depends on the square of the number of samples times the cost for the classifier generation and the classifying step. On the other hand it gives the most reliable prediction of the performance of a classifier of all hold-out methods.

# 5 Classifier Generation

By now the consideration of the feature selection and error estimation had nothing to do with the model that we may choose for the classifier. Many options exist for the system that later on tries to identify correctly an unknown sample [8]. Parametric or unparametric methods, neural networks or fuzzy set inference could all be used for the classification job.

Reminding the context of this work, we choose a nonparametric model since no assumptions should be made of the sensor data that has to be analyzed.

## 5.1 Nearest Neighbor Classifier for Error Estimation

The simplest nonparametric classifier technique is used for the purpose of error estimation. An unknown sample is said to belong to a certain class $c$ if its $k$ nearest neighbors in the multidimensional Euclidean space belong to the same class (K-Nearest-Neighbor). “Near” implies that a distance measure is available which gives a continuous numerical value for two samples and satisfies the distance properties. Usually the Euclidean norm is chosen as the distance measure. Hence for two samples $\underline{x}_i$ and $\underline{x}_j$ the distance is defined as $d(\underline{x}_i, \underline{x}_j) = \|\underline{x}_i - \underline{x}_j\|$.

We furthermore do not perform any data reduction over the training set. All samples are considered as prototypes [9]. This keeps the error estimation free of negative influences of the training set compression step (data to knowledge), but is more expensive in calculus because an unknown sample must be compared to *all* training samples.

## 5.2 Learning Vector Quantizer

What has been achieved so far is a reasonable dimension of the sensor’s feature value vectors. This can be considered as a horizontal data reduction. If the samples have to be found useful for classification a vertical data reduction has to be performed as well, i.e. the reduction of all labeled class members to a small amount of representative prototypes. Also for this task a general purpose method must be employed, since normally the supervision problem is not known a priori. The statistical distribution of the classes is in general multimodal. That means that the samples of the same class can be spread in the multidimensional feature space in a quite arbitrary manner. Parametric techniques for the classifier model, like the multivariate normal distribution bear the risk of very imprecise estimates for the class boundaries. In a family of problems where it is difficult to estimate system parameters a non-parametric technique is highly recommended.

We use the work of Kohonen [1], [2], [3] to solve that general classification problem. His self-organizing map in its supervised form is able to perform a learning of a variety of identifier problems. The labeled samples are fed into a generalization module which tries to find a set of “codebook vectors” which represent the knowledge about the class membership of the samples. Different learning procedures can be selected to produce the codebook vectors. They are all members of Vector Quantizers (VQ) which iteratively seek values for the codebook vectors which minimize a reconstruction error of the sample set.

What makes this form of a classifier especially attractive for monitoring and supervision tasks is the fact that no a priori knowledge must be available and the classification is very fast. The designer of the classifier simply applies a vector quantizing over the samples with the reduced feature dimension and obtains a set of codebook prototypes to which all unknown feature vectors are compared in the later operation phase. This permits a on-line real time supervision of the production process.

# 6 Visualization of Multidimensional Sensor Data

Multidimensional data is impossible to visualize if the dimension is higher than 2 or 3 at most. A point in a 17-dimensional feature space which is far apart from another point in the same space could be identical is a subspace, e.g. in the 2-dimensional subspace of its first two components. The algorithm of Sammon [4] calculates the mutual multidimensional distance between the samples and arranges the samples iteratively in the x-y-plane. Thus it permits to get an idea of how far samples are apart among them. One can expect that classes that distinguish well in a higher dimensional space maintain this quality also in a lower dimension. The main objective of this module is to provide the supervision system designer with information about the quantitative and qualitative nature of the sensor data. It could e.g. lead to the conclusion that the sound sensor is very appropriate for the detection of malfunctioning during the drilling of a hole into a metal block or that the sensor for the current should always deliver the same values for all production situations, independent of the task. Otherwise a fault is highly probable.

As a small additional module, linear correlation between two different features can be calculated and visualized. This can be done for one or more classes.

# 7 Implementation

All concepts presented so far have been programmed in a highly portable "C"-code. The toolbox was compiled on a variety of UNIX platforms and on a MS-DOS system.

The data acquisition is performed via files in a predefined format. The feature selection and error estimation is implemented in a newly developed part. The learning of the codebook vectors is done by using the program package "LVQ_PAK" [3] developed by the group of Kohonen. The tool provides an interface to that program. Included into this package is the utility program which performs the Sammon mapping. Our own work interfaces this module to a powerful plotting package: "GNUPLOT". It is now possible to accumulate data, select features, visualize the data and generating the classifier on-line. This is the actual state of the art of the package. Future extensions will orient themselves at the problems that arise during further investigation of monitoring and supervision.

# 8 Case Study

As a test bed we are using the data measured from various types of sensors which monitor a NC machine. The DENDFORD STARTURN 4 lathe machine is a semi-professional workbench to produce parts that are normally designed in a CAD environment. The sensors during the working phase are: sound, axis movement vibrations and mains current consumption. The sensor data can be transferred on-line to a UNIX platform. There the data analysis toolbox enables the further processing. See also [11] in this proceedings for a more detailed description of the test bed.

Results are presented which try to attribute quality labels to certain sensors for particular working conditions.

## 8.1 Example I: Sound Sensor for Spindle

With the axes stopped, the spindle speed was exercised for 3 different rotations: 500rpm, 600rpm and 700rpm. The sound of the rotating spindle was sampled and the signal buffer was transformed to the frequency domain by a Fourier transform. This yielded 256 frequencies which are considered as the features of the 3 rotation classes. A small number of samples was taken, namely 16, 13 and 13 for the 3 rotation classes in order to have better visualization conditions (more samples showed the same class distribution characteristics).

### 8.1.1 Feature Selection

The following table shows the results of three features selection strategies of chapter 3. From the total of 256 the best 10 features for each strategy were selected.

TABLE I
FEATURE SELECTION AND ERROR ESTIMATION

| Classes | Nr. 1 "b500m3" | | | Nr. 2 "b600m3" | | | Nr. 3 "b700m3" | | |
|---|---|---|---|---|---|---|---|---|---|
| Nr. of samples | 16 | | | 13 | | | 13 | | |
| Selection Strategy | | | | | | | | | |
| | *Univar* | | | *M-Covar* | | | *MinErr* | | |
| Selected feature Nr. | Frequency Nr. | Criterion | Estimated error [%] | Frequency Nr. | Criterion | Estimated error [%] | Frequency Nr. | Criterion | Estimated error [%] |
| 1 | 47 | -1.74 | 57.14 | 147 | 20.31 | 45.24 | 107 | 33.33 | = Criterion |
| 2 | 156 | -1.80 | 23.81 | 44 | 12.73 | 38.10 | 77 | 16.67 | |
| 3 | 109 | -1.96 | 23.81 | 204 | 9.20 | 47.62 | 125 | 16.67 | |
| 4 | 137 | -2.00 | 19.05 | 137 | 9.18 | 40.48 | 201 | 11.90 | |
| 5 | 157 | -2.02 | 14.29 | 107 | 20.52 | 19.05 | 58 | 4.76 | |
| 6 | 147 | -2.05 | 14.29 | 246 | 17.68 | 21.43 | 59 | 4.76 | |
| 7 | 107 | -2.07 | 2.38 | 255 | 12.40 | 21.43 | 69 | 4.76 | |
| 8 | 241 | -2.16 | 2.38 | 47 | 9.53 | 4.76 | 14 | 4.76 | |
| 9 | 108 | -2.45 | 7.14 | 119 | 10.84 | 7.14 | 1 | 4.76 | |
| 10 | 192 | -2.65 | 7.14 | 202 | 7.83 | 4.76 | 2 | 4.76 | |

The table can be interpreted in the following manner: Using e.g. the feature selection strategy M-Covar, the selection of feature number 5 gave for the 107th of all 256 possible features the highest selection criterion of 20.52. Then the error estimation with the leave-one-out previewed an error of 19.5% for a classifier that was based on the first 5 features.

# 9 Conclusions and Future Work

In this paper we presented a software toolbox for the analysis and visualization of sensor data in the context of machine tool supervision. The tool is conceived for the characterization and classification of machine states which are defined by multidimensional continuous feature vectors. The main objective is the selection of the most discriminant subset of the features that are provided. Error estimation of a classifier based on these features can be performed. Multidimensional data can be visualized by a mapping to only two dimensions.

The toolbox makes as few assumptions of the available data as possible. An interface to a class learning module is provided.

# Annex

## A.1 Software sources

The program source together with the documentation can be obtained by remote file transfer:

```
unix> ftp ftp.fct.unl.pt
Name: anonymous
ftp> cd pub/di/packages
ftp> get tooldiag.README
ftp> get tooldiag.tar.Z
ftp> quit
```

## A.2 Fisher's Iris flower data set

An interesting study is the effect of the sensor analysis toolbox on the classical data set of Fisher [10]. A set of 150 samples of feature dimension 4 describes 3 different flower classes, 50 samples per class. The particular feature values are explicitly shown in TABLE II. In TABLE III the results for the feature selection and error estimation are presented. It is interesting to notice that all three strategies select almost the same set of features. *Qualitatively* all strategies state that the two features "petal length" and "petal width" are *good* features; the reduce the estimated classification error considerably. The other two features "sepal length" and "sepal width" do not contribute to notable improvements in the error rate. They are *bad* features.

Figure 1 shows the visualization of the data set by the algorithm of Sammon.

This example of data analysis emphasizes the philosophy of the toolbox. The use of different strategies leads to similar results. It permits conclusions about the quality of the features, and if it makes sense at all to use these features to classify the three flower species.

TABLE II
IRIS FLOWER DATA SET

| | Species | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Iris setosa | | | | Iris versicolor | | | | Iris virginica | | | |
| | Features | | | | Features | | | | Features | | | |
| Sample Nr. | sepal length | sepal width | petal length | petal width | sepal length | sepal width | petal length | petal width | sepal length | sepal width | petal length | petal width |
| 1 | 5.1 | 3.5 | 1.4 | 0.2 | 7.0 | 3.2 | 4.7 | 1.4 | 6.3 | 3.3 | 6.0 | 2.5 |
| 2 | 4.9 | 3.0 | 1.4 | 0.2 | 6.4 | 3.2 | 4.5 | 1.5 | 5.8 | 2.7 | 5.1 | 1.9 |
| 3 | 4.7 | 3.2 | 1.3 | 0.2 | 6.9 | 3.1 | 4.9 | 1.5 | 7.1 | 3.0 | 5.9 | 2.1 |
| 4 | 4.6 | 3.1 | 1.5 | 0.2 | 5.5 | 2.3 | 4.0 | 1.3 | 6.3 | 2.9 | 5.6 | 1.8 |
| 5 | 5.0 | 3.6 | 1.4 | 0.2 | 6.5 | 2.8 | 4.6 | 1.5 | 6.5 | 3.0 | 5.8 | 2.2 |
| 6 | 5.4 | 3.9 | 1.7 | 0.4 | 5.7 | 2.8 | 4.5 | 1.3 | 7.6 | 3.0 | 6.6 | 2.1 |
| 7 | 4.6 | 3.4 | 1.4 | 0.3 | 6.3 | 3.3 | 4.7 | 1.6 | 4.9 | 2.5 | 4.5 | 1.7 |
| 8 | 5.0 | 3.4 | 1.5 | 0.2 | 4.9 | 2.4 | 3.3 | 1.0 | 7.3 | 2.9 | 6.3 | 1.8 |
| 9 | 4.4 | 2.9 | 1.4 | 0.2 | 6.6 | 2.9 | 4.6 | 1.3 | 6.7 | 2.5 | 5.8 | 1.8 |
| 10 | 4.9 | 3.1 | 1.5 | 0.1 | 5.2 | 2.7 | 3.9 | 1.4 | 7.2 | 3.6 | 6.1 | 2.5 |
| 11 | 5.4 | 3.7 | 1.5 | 0.2 | 5.0 | 2.0 | 3.5 | 1.0 | 6.5 | 3.2 | 5.1 | 2.0 |
| 12 | 4.8 | 3.4 | 1.6 | 0.2 | 5.9 | 3.0 | 4.2 | 1.5 | 6.4 | 2.7 | 5.3 | 1.9 |
| 13 | 4.8 | 3.0 | 1.4 | 0.1 | 6.0 | 2.2 | 4.0 | 1.0 | 6.8 | 3.0 | 5.5 | 2.1 |
| 14 | 4.3 | 3.0 | 1.1 | 0.1 | 6.1 | 2.9 | 4.7 | 1.4 | 5.7 | 2.5 | 5.0 | 2.0 |
| 15 | 5.8 | 4.0 | 1.2 | 0.2 | 5.6 | 2.9 | 3.6 | 1.3 | 5.8 | 2.8 | 5.1 | 2.4 |
| 16 | 5.7 | 4.4 | 1.5 | 0.4 | 6.7 | 3.1 | 4.4 | 1.4 | 6.4 | 3.2 | 5.3 | 2.3 |
| 17 | 5.4 | 3.9 | 1.3 | 0.4 | 5.6 | 3.0 | 4.5 | 1.5 | 6.5 | 3.0 | 5.5 | 1.8 |
| 18 | 5.1 | 3.5 | 1.4 | 0.3 | 5.8 | 2.7 | 4.1 | 1.0 | 7.7 | 3.8 | 6.7 | 2.2 |
| 19 | 5.7 | 3.8 | 1.7 | 0.3 | 6.2 | 2.2 | 4.5 | 1.5 | 7.7 | 2.6 | 6.9 | 2.3 |
| 20 | 5.1 | 3.8 | 1.5 | 0.3 | 5.6 | 2.5 | 3.9 | 1.1 | 6.0 | 2.2 | 5.0 | 1.5 |
| 21 | 5.4 | 3.4 | 1.7 | 0.2 | 5.9 | 3.2 | 4.8 | 1.8 | 6.9 | 3.2 | 5.7 | 2.3 |
| 22 | 5.1 | 3.7 | 1.5 | 0.4 | 6.1 | 2.8 | 4.0 | 1.3 | 5.6 | 2.8 | 4.9 | 2.0 |
| 23 | 4.6 | 3.6 | 1.0 | 0.2 | 6.3 | 2.5 | 4.9 | 1.5 | 7.7 | 2.8 | 6.7 | 2.0 |
| 24 | 5.1 | 3.3 | 1.7 | 0.5 | 6.1 | 2.8 | 4.7 | 1.2 | 6.3 | 2.7 | 4.9 | 1.8 |
| 25 | 4.8 | 3.4 | 1.9 | 0.2 | 6.4 | 2.9 | 4.3 | 1.3 | 6.7 | 3.3 | 5.7 | 2.1 |
| 26 | 5.0 | 3.0 | 1.6 | 0.2 | 6.6 | 3.0 | 4.4 | 1.4 | 7.2 | 3.2 | 6.0 | 1.8 |
| 27 | 5.0 | 3.4 | 1.6 | 0.4 | 6.8 | 2.8 | 4.8 | 1.4 | 6.2 | 2.8 | 4.8 | 1.8 |
| 28 | 5.2 | 3.5 | 1.5 | 0.2 | 6.7 | 3.0 | 5.0 | 1.7 | 6.1 | 3.0 | 4.9 | 1.8 |
| 29 | 5.2 | 3.4 | 1.4 | 0.2 | 6.0 | 2.9 | 4.5 | 1.5 | 6.4 | 2.8 | 5.6 | 2.1 |
| 30 | 4.7 | 3.2 | 1.6 | 0.2 | 5.7 | 2.6 | 3.5 | 1.0 | 7.2 | 3.0 | 5.8 | 1.6 |
| 31 | 4.8 | 3.1 | 1.6 | 0.2 | 5.5 | 2.4 | 3.8 | 1.1 | 7.4 | 2.8 | 6.1 | 1.9 |
| 32 | 5.4 | 3.4 | 1.5 | 0.4 | 5.5 | 2.4 | 3.7 | 1.0 | 7.9 | 3.8 | 6.4 | 2.0 |
| 33 | 5.2 | 4.1 | 1.5 | 0.1 | 5.8 | 2.7 | 3.9 | 1.2 | 6.4 | 2.8 | 5.6 | 2.2 |
| 34 | 5.5 | 4.2 | 1.4 | 0.2 | 6.0 | 2.7 | 5.1 | 1.6 | 6.3 | 2.8 | 5.1 | 1.5 |
| 35 | 4.9 | 3.1 | 1.5 | 0.2 | 5.4 | 3.0 | 4.5 | 1.5 | 6.1 | 2.6 | 5.6 | 1.4 |
| 36 | 5.0 | 3.2 | 1.2 | 0.2 | 6.0 | 3.4 | 4.5 | 1.6 | 7.7 | 3.0 | 6.1 | 2.3 |
| 37 | 5.5 | 3.5 | 1.3 | 0.2 | 6.7 | 3.1 | 4.7 | 1.5 | 6.3 | 3.4 | 5.6 | 2.4 |
| 38 | 4.9 | 3.6 | 1.4 | 0.1 | 6.3 | 2.3 | 4.4 | 1.3 | 6.4 | 3.1 | 5.5 | 1.8 |
| 39 | 4.4 | 3.0 | 1.3 | 0.2 | 5.6 | 3.0 | 4.1 | 1.3 | 6.0 | 3.0 | 4.8 | 1.8 |
| 40 | 5.1 | 3.4 | 1.5 | 0.2 | 5.5 | 2.5 | 4.0 | 1.3 | 6.9 | 3.1 | 5.4 | 2.1 |
| 41 | 5.0 | 3.5 | 1.3 | 0.3 | 5.5 | 2.6 | 4.4 | 1.2 | 6.7 | 3.1 | 5.6 | 2.4 |
| 42 | 4.5 | 2.3 | 1.3 | 0.3 | 6.1 | 3.0 | 4.6 | 1.4 | 6.9 | 3.1 | 5.1 | 2.3 |
| 43 | 4.4 | 3.2 | 1.3 | 0.2 | 5.8 | 2.6 | 4.0 | 1.2 | 5.8 | 2.7 | 5.1 | 1.9 |
| 44 | 5.0 | 3.5 | 1.6 | 0.6 | 5.0 | 2.3 | 3.3 | 1.0 | 6.8 | 3.2 | 5.9 | 2.3 |
| 45 | 5.1 | 3.8 | 1.9 | 0.4 | 5.6 | 2.7 | 4.2 | 1.3 | 6.7 | 3.3 | 5.7 | 2.5 |
| 46 | 4.8 | 3.0 | 1.4 | 0.3 | 5.7 | 3.0 | 4.2 | 1.2 | 6.7 | 3.0 | 5.2 | 2.3 |
| 47 | 5.1 | 3.8 | 1.6 | 0.2 | 5.7 | 2.9 | 4.2 | 1.3 | 6.3 | 2.5 | 5.0 | 1.9 |
| 48 | 4.6 | 3.2 | 1.4 | 0.2 | 6.2 | 2.9 | 4.3 | 1.3 | 6.5 | 3.0 | 5.2 | 2.0 |
| 49 | 5.3 | 3.7 | 1.5 | 0.2 | 5.1 | 2.5 | 3.0 | 1.1 | 6.2 | 3.4 | 5.4 | 2.3 |
| 50 | 5.0 | 3.3 | 1.4 | 0.2 | 5.7 | 2.8 | 4.1 | 1.3 | 5.9 | 3.0 | 5.1 | 1.8 |

TABLE III
FEATURE SELECTION AND ERROR ESTIMATION

| Classes | Nr. 1 "setosa" | | | Nr. 2 "virginica" | | | Nr. 3 "versicolor" | | |
|---|---|---|---|---|---|---|---|---|---|
| Nr. of samples | 50 | | | 50 | | | 50 | | |
| | Selection Strategy | | | | | | | | |
| | *Univar* | | | *M-Covar* | | | *MinErr* | | |
| Selected feature Nr. | Frequency Nr. | Criterion | Estimated error [%] | Frequency Nr. | Criterion | Estimated error [%] | Frequency Nr. | Criterion | Estimated error [%] |
| 1 | 3 | 0.81 | 12.00 | 3 | 107.2 | 12.00 | 2 | 12.00 | = Criterion |
| 2 | 2 | 0.77 | 4.67 | 2 | 25.65 | 4.67 | 3 | 4.67 | |
| 3 | 0 | -0.43 | 4.67 | 0 | 11.66 | 4.67 | 0 | 4.67 | |
| 4 | 1 | -3.32 | 4.00 | 1 | 4.07 | 4.00 | 1 | 4.00 | |

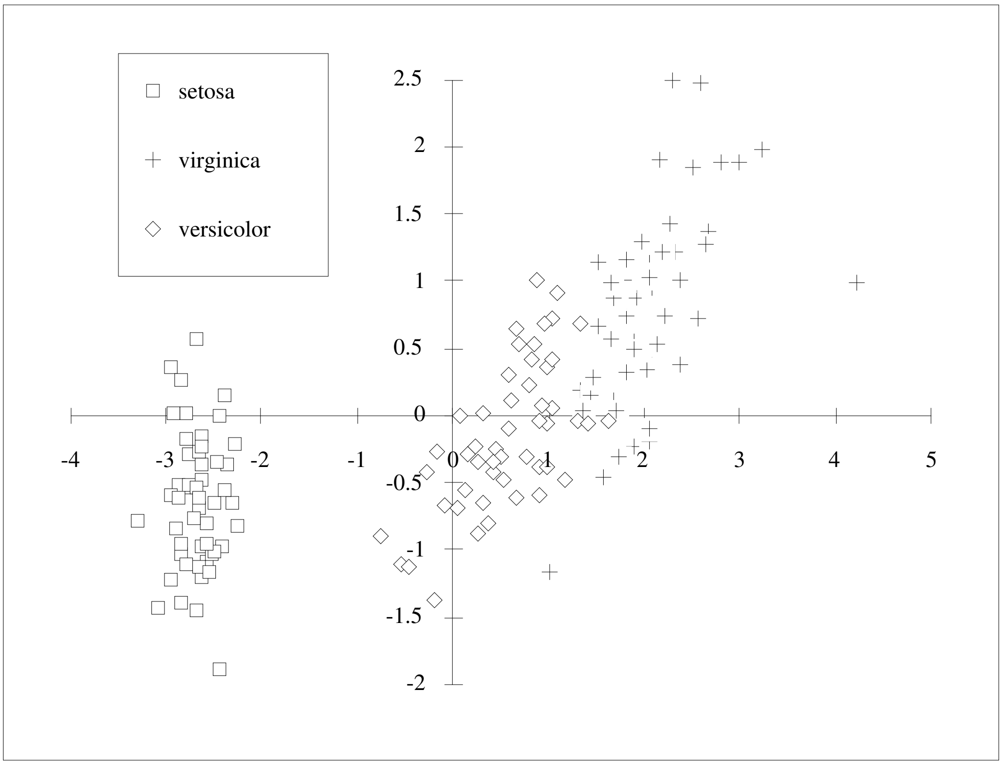


Figure 2: *Sammon plot of Fisher's iris flower data base. The original 4-dimensional feature vector was mapped to 2 dimensions by a non-linear mapping method. The number of iterations was set to 100.*

# References


[1] T. Kohonen, "The self-organizing map," in Proc. of the IEEE, vol. 78, no. 9, pp. 1464-1480, Sept. 1990.

[2] ——, 1986. "Learning Vector Quantization," Helsinki University of Technology, Lab. of Comp. and Information Science, Report TKK-F-A-601.

[3] ——, et al., "LVQ_PAK, Vers. 2.0", Technical Manual, Helsinki Univ. of Technology, Lab. of Comp. and Inf. Science, Jan. 1992.

[4] J. W. Sammon Jr., "A nonlinear mapping for data structure analysis", IEEE Trans. on Computers, C-18(5): 401-409, May 1969.

[5] P. A. Devijver and J. Kittler, "Pattern Recognition: A Statistical Approach," Prentice/Hall Int., London, 1982.

[6] D. Guinea, A. Ruiz, and L.J. Barrios, "Multi-sensor integration — An automatic feature selection and state identification methodology for tool wear estimation," in *Computers in Industry,* Volume 17, Nr. 2&3, L. Monostori (Ed.), pp. 121-130, Elsevier, Nov. 1991.

[7] W. W. Cooley and P. R. Lohnes, "Multivariate Data Analysis," New York: Wiley, 1971.

[8] Duda, R.O. and Hart, P.E. 1973. "Pattern classification and scene analysis," Wiley, New York.

[9] Tou, J.T. and Gonzalez, R.C. 1974. "Pattern recognition principles," Reading: Addison-Wesley.

[10] Fisher, R. 1936. "The use of multiple measurements in taxonomic problems," in Annals of Eugenics. 7, 179-188.

[11] Barata, M. M., Rauber T. W., and Steiger-Garção A., "Prognostic and Monitoring System for CNC Machines," in Proc. Tooldiag 93-Int. Conf. on Fault Diagnosis, Toulouse, France, to be published, April 1993.