Statistical Classifier with Ordered Decisions as an Image Based Controller with Application to Gas Burners

Ewaryst Rafajłowicz[1], Halina Pawlak-Kruczek[2], and Wojciech Rafajłowicz[1]

[1] Institute of Computer Engineering control and Robotics
[2] Institute of Heat Engineering and Fluid Mechanics,
Wrocław University of Technology Wrocław, Poland
ewaryst.rafajlowicz@pwr.wroc.pl

Abstract. We consider a statistical decision problem as a tool for solving control problems with a camera in the loop. The first stage is features extraction from images. Its role is to process images in order to extract features relevant for the control problem. Then, they are fed as inputs to the Bayesian decision problem. At the second stage a loss function, which is a sum of squared deviations of decisions from true decisions is considered. Finally, an approximation of the optimal decision rule is proposed, using a learning sequence of decisions, which – together with feature extracting algorithms – form the control system. The proposed approach is illustrated by a system that is dedicated to control natural gas burners.

Keywords: pattern recognition, ordered labels, control, image processing, flame, burner.

1 Introduction

A camera in a control loop is a very rich source of information, in many cases even too rich. In such a case a control system has to reduce information content. Otherwise, very similar images are considered as different, which may lead to the instability of the control loop. It seems that the Bayes decision theory is a good source of general ideas on how to select proper decisions when an environment is randomly changing (see [5] for fundamentals of general Bayesian decision theory) and [4] or [7] for more specialized pattern recognition setting. However, we have to point out that the Bayes decision theory has two points, which are not compatible with applications in control theory. Namely,

1. the Bayes decision theory is static in the sense that the decision is taken only once (an idea of using classifiers in a nonstationary environment has been proposed in [18], [19]),
2. it assumes that we know all necessary a priori probabilities of an environment states and all probability density functions of states from each class.

L. Rutkowski et al. (Eds.): ICAISC 2014, Part I, LNAI 8467, pp. 586–597, 2014.

In order to apply the Bayesian approach we have to start from assumptions made in 2), but then we have to take a more realistic empirical approach that is based on estimating these quantities from observations. Concerning 1), we have to take into account that

- a decision can be changed when the next image is collected and processed,
- possible reasons for changing decisions include:
  - disturbances in the process under control,
  - a correction of the previous decision, even when disturbances were not present,
  - erroneously interpreted previous frames, resulting in incorrect decisions.
- processing of images should be sufficiently fast in order to ensure proper behavior of the controlled process.

The last point, in turn, requires the extraction of features from images that are relevant for process control. Thus, we arrive to the two stage approach. Namely,

1st stage consists of image acquisition and extraction of relevant features,
2nd stage consists of calculating decisions.

In this paper we assume that the process under control has no long memory. In other words, its dynamics (transient process) can be neglected. If it cannot be ignored, then the proposed approach can be extended, e.g., by assuming that states of the process form a Markov chain, but this generalization is outside the scope of this paper.

We also assume that for a proper process control it suffices to use a finite number, say $l > 1$, of decisions. In other words, we quantize a decision process. The class of decisions can be arbitrarily large (in order to approximate an arbitrary decision function), but it must be finite or at most countable.

In opposite to the classical setting of Bayes problems, in which labels of the decisions are arbitrary and unordered, we consider the problem with decision labels that are linearly ordered. There are practical reasons to take the above point of view. Namely, when decisions are applied to the process under control and we admit some errors in decisions, it is important to attach a larger loss when instead of decision "1" we take decision "4" then if decision "2" (closer to "1") is undertaken. We can assume that $\mathcal{I} \overset{def}{=} \{1, 2, \ldots, I\}$, $I \geq 2$ is the set of labels of our decisions. In practice, decision "1" may mean, apply a force of 10 N to a controlled system, while decisions "2" and "3" correspond to applying forces 15 N and 22 N, respectively. It is only of importance, that 10 N is closer to 15 N, then to 22 N.

As an optimality criterion we take the expectation of the sum of differences between the "true" (mostly desirable) decision label and our decision. We shall prove that the optimal decision rule is extremely simple in this case. Namely, its the a posteriori mean (possibly "rounded" to the nearest decision from the above list).

As the next step, we discuss an empirical version of the controller, i.e., we propose an algorithm, which approximates the optimal decision rule.

It should be mentioned that different orderings were suggested in [20], [13], but they were introduced in the feature space, while here, we consider ordering in the space of labels. In [15] two dimensional class labels were considered and applied to a fault location problem.

2 Decision Problem – Theoretical Setting

Let $Y(t)$ denote an image of the process at time t. $Y(t)$ is an $r \times c$ matrix, where r is the total number of pixels that are stacked in a vector, c is the number of colors, e.g., three in RGB format. Denote by Ω an operator (linear or nonlinear) defined on the above described set of matrices with values in R^d, i.e., $X(t) = \Omega(Y(t))$ is d-dimensional vector. We shall interpret operator Ω as a feature extractor which extracts features relevant to the controlled process and forms a pattern $X(t)$ that is sufficient for proper decision making. The form of operator Ω is problem dependent and not discussed in this section.

2.1 Assumptions

1) For each t, $X(t) \in R^d$ is a random vector, for which a correct decision, denoted as $i(t)$, is an element of I, labeled as $1, 2, \ldots, I$.

2) New pair (X, i) is a random vector representing features and a correct decision i, which is unknown for a new feature vector X to be classified.

3) Probability distribution of (X, i) is unknown, but we also have a learning sequence

$$(X^{(k)} = X(t_k), i^{(k)} = i(t_k)), \quad k = 1, 2, \ldots, n$$

of observed $X^{(k)} \in R^d$ and the corresponding correct decisions $i^{(k)} \in \{1, 2, \ldots, I\}$. We assume that $(X^{(k)}, i^{(k)})$'s are independent, identically distributed random vectors with the same probability distribution as (X, i).

4) Denote by $0 \leq q(i) \leq 1$, a priori probability that for X a correct decision is i, $i = 1, 2, \ldots, I$, $\sum_{i=1}^{I} q(i) = 1$.
5) The next ingredient of the problem setting is a loss function, $L(i, j)$ say, which attaches loss $L(i, j)$ if the correct decision for X is i, while our decision $j \in I$. In this paper we take

$$L(i,\, j) = (i - j)^2, \quad i,\, j \in \mathcal{I} \tag{1}$$

as the loss function. It reflects our idea that it is reasonable to attach a larger loss when incorrect decision j is far from the correct one, i.e., i.

Let E_X denotes the expectation w.r.t. X, while $P(i|X)$ is the a posteriori probability that for new feature vector X the correct decision is i. In other words, $P(i|X = x)$ is the conditional probability of the event that i is the correct label of the decision for given vector of features $X = x$.

2.2 Problem Statement

The aim is to find a decision function $\Psi(X)$, which specifies a decision from I for X and such that it minimizes the expected loss given by:

$$R(\Psi) \;=\; E_X\left[\sum_{i=1}^{I}\left(i-\Psi(X)\right)^2\;P(i|X)\right], \tag{2}$$

In other words, our aim is to minimize the risk $R(\Psi)$, which is our loss that is averaged of all possible feature vectors X, while averaging is done according to the probability distribution of X. In order to ensure correctness of (2), we have to assume that we are looking for minimizer $\Psi^*(x)$, say, in the class of measurable functions.

The above problem statement is a theoretical one, because we do not know $P(i|X)$'s and probability distribution of X. However, it is reasonable to firstly solve problem (2) as if these quantities were known (see the next Section) and then, to estimate $\Psi^*(x)$ from the sequence of past correct decisions. This approach can be named "learning statistically optimal control".

3 From Image Features to Optimal Decision Rule

The outer expectation in (2) is carried out with respect to a nonnegative probability measure corresponding to the distribution of X. Thus, it suffices to minimize the expression in the square brackets in (2) for each vector of feature X separately. In other words, in order to minimize $R(\Psi)$ it suffices to minimize the conditional risk, which is defined as follows

$$r(\psi, x) \;\overset{def}{=}\; \sum_{i=1}^{I}\left(i-\psi\right)^2\;P(i|X=x) \tag{3}$$

with respect to ψ, which is a real variable, while x is treated as a parameter. Notice that ψ^* which minimizes (3) must depend on x and this dependence is in fact our decision making unit Ψ^*, that can be also described as follow:

$$\Psi^*(x) \;=\; \arg\min_{\psi}\; r(\psi, x) \tag{4}$$

for all $x \in R^d$ in the range of X.

Note that minimizing (4) we have to ensure that $\Psi^*(x)$ is a positive integer form the class of admissible decisions I.

3.1 Optimal Decision Rule

The decision rule $\Psi^*(x)$, which minimizes (3), can be derived in two steps that follows.

Step 1. For fixed x we have to minimize the sum of $(i-\psi)^2$ with weights, which sum up to 1. The best ψ, which is further denoted as $\tilde{\Psi}(x)$, is the a posteriori mean:

$$\tilde{\Psi}(x) \;=\; \sum_{i=1}^{I} i\,P(i|X=x) \tag{5}$$

It is worth explaining how formula (5) works in an idealized case. If the range of random vectors (features) X can be divided into disjoint subregions Γ_i, each corresponding to unique decision i, then $P(i|X=x)=1$ exactly for $x\in\Gamma_i$ and $P(j|X=x)=0$ for all $j\neq i$. In this case $\tilde{\Psi}(x)=i$ for $x\in\Gamma_i$.

In more realistic cases we have to perform the following step.

Step 2. In order to ensure that our decision is a positive integer we round it as follows:

$$\Psi^*(x) \;=\; \text{ROUND}\left(\tilde{\Psi}(x)\right), \tag{6}$$

where ROUND(t) is an integer, which is closest to t.

Let us assume the existence of probability densities $f(x|i)$, which describes the conditional p.d.f. of X, provided that it corresponds to i-th decision from I. Then, according to the Bayes rule, $P(i|X=x)$ is given by

$$P(i|X=x) \;=\; \frac{f(x|i)\,q(i)}{f(x)}, \quad i=1,2,\ldots,I, \tag{7}$$

$$f(x) \;\overset{def}{=}\; \sum_{l=1}^{I} f(x|l)\,q(l)\ , \tag{8}$$

which allows us to express the a posteriori probabilities in terms of probability densities corresponding to i-th and all other decisions and a priori probabilities that i-th decision is appropriate. Reasoning analogously as in the proof of Theorem 1 in [16] the following result can be derived.

Theorem 1. *For linearly ordered decision labels the expected loss (2) is minimized by decision rule (5), (6).*

3.2 Countable Decision Set and Exponential Family of Features

Here we illustrate the above theoretical derivations and simultaneously we shall show that the above theory easily conveys to an infinite but countable set of decisions.

Assume (in this section only) that we have only one feature $x\in[0,\infty)$. Probability densities have the form:

$$f(x|i) = n\,\frac{x^k}{k!}\,exp(-n\,x), \quad x\in[0,\infty), \quad i=0,\,1,\ldots\ , \tag{9}$$

where $n\geq 1$ is a selected integer. For convenience we have labeled classes from 0. We also assume that a priori probabilities that i-th decision is the proper one are equal to $q_i=(1-q)\,q^i$, $i=0,\,1,\ldots$, $0<q<1$. Then,

$$f(x) = \sum_{i=0}^{\infty} f(x|i)\, q_i = n(q-1)\left(-e^{n(q-1)x}\right) \tag{10}$$

while

$$\sum_{i=0}^{\infty} i\, f(x|i)\, q_i = n^2\left(q^2-q\right) x \left(-e^{n(q-1)x}\right) \tag{11}$$

Thus, we obtain the optimal decision rule $\Psi^*(x) = \mathrm{ROUND}(n\,q\,x)$.

4 Second Stage – Toward Empirically Tuned Controller

When $P(i|X=x)$ and q_i, $i \in I$ are unknown, one can not apply Thm. 1 directly. The well-established way of circumventing this difficulty consists of the following steps:

1. Collect a learning sequence

$$\mathcal{L}_n = (X^{(k)}, i^{(k)}), \quad k = 1, 2, \ldots, n$$

that contains pairs: a feature vector $X^{(k)}$ and the corresponding proper decision label $i^{(k)}$ (suggested by experts, which are assumed to provide proper decisions).

2. Estimate $P(i|X=x)$ and q_i, $i \in I$ from $\mathcal{L}_n$
3. Insert the above-mentioned estimates into the optimal decision rule (5), (6) and use it as an approximation of the optimal one.

The above way frequently leads to empirical decision rules that are asymptotically (as $n \to \infty$) as good as the optimal one (in the sense of the expected loss).

Estimators of $f(x|i)$ can be based on orthogonal expansions or the well-known Parzen-Rosenblatt kernel estimator, partitioning estimators and many others (see [6] and [7] for detailed discussions and extensive bibliographies). All these approaches operate directly on the whole learning sequence, which has to be stored and available for each pattern to be recognized. This feature is highly undesirable when one needs to use an empirical decision rule on-line for a process control.

4.1 Empirical Decision Rule

Therefore, we propose simplified estimators of $f(x|i)$'s that do not require the storage of the whole learning sequence in order to make current decision. This goal can be achieved by applying the well known neural nets with radial basis functions.

Denote by $K(t) \geq 0$, $t \in R$ a kernel, which is such that

$$\int_{-\infty}^{\infty} K(t) = 1, \quad \int_{-\infty}^{\infty} t\, K(t) = 0, \quad \int_{-\infty}^{\infty} t^2\, K(t) < \infty$$

Define

$$\mathcal{I}(i) = \{(X^{(k)}, i^{(k)}) : i^{(k)} = i\}.$$

$$\sum_{i=1}^{I} Card(\mathcal{I}(i)) = n,$$

where $Card$ denotes the cardinality of a set. Let $n(i) = Card(\mathcal{I}(i))$ denote the number of decisions with the label i in the learning sequence.

The simplest task is estimating a priori probability that a decision has label i. It suffices to set

$$\hat{q}(i) = \frac{n(i)}{n}, i = 1, 2, \ldots, I \tag{12}$$

as the estimator of $q(i)$.

As estimators $\hat{f}(x|i)$ of $f(x|i)$ we take the following RBF net:

$$\hat{f}(x|i) = \frac{1}{J(i)\, h^d(n(i), i)} \sum_{j=1}^{J(i)} K\left(\frac{||x - C_i^{(j)}||}{h(n(i), i)}\right), \quad i = 1,\, 2, \ldots,\, I, \tag{13}$$

where
– $C_i^{(j)} \in R^d$, $j = 1,\, 2, \ldots,\, J(i)$ are centers of RBF's for estimating $f(x|i)$,
– $h(n(i), i)$ is i-th smoothing parameter that depends on $n(i)$,
– $||.||$ is a norm in R^d.

We refer the reader to rich bibliography and recent results on RBF's contained in [2], [4], [15], [16], [18], [19], [21], [22], [13], [10]. Define the second, composite, RBF net as follows:

$$\hat{f}(x) \;=\; \sum_{i=1}^{I} \hat{q}_i\, \hat{f}(x|i). \tag{14}$$

It consists of RBF nets (13) with weights $\hat{q}_i$.

In (13) and (14) the role of the learning sequence is not directly visible. It is "hidden" in selecting centers $C_i^{(j)} \in R^d$. Firstly, we select those $(X^{(k)},\, i^{(k)})$'s that are contained in $\mathcal{I}(i)$ for given i. Then, they serve for selecting $C_i^{(j)}$ $j = 1,\, 2, \ldots J(i)$ for the same i. We omit the discussion on selecting centers, since it is well covered in the available literature.

Define the estimators $\hat{p}(i, x)$ of $P(i|X = x)$ as follows

$$\hat{p}(i, x) \;=\; \frac{\hat{q}_i\, \hat{f}(x|i)}{\hat{f}(x)}, \tag{15}$$

Now, as the empirical decision rule $\hat{\Psi}$ we take

$$\hat{\Psi}(x) \;=\; \text{ROUND}\left[\frac{1}{\hat{f}(x)} \sum_{i=1}^{I} i\, \hat{q}_i\, \hat{f}(x|i)\right]. \tag{16}$$

4.2 Important Special Case

Training controllers (16) with $\hat{f}(x|i)$ given by (13) usually requires a long learning sequence. In many cases one can simplify (13) by setting $J(i) = 1$, $i = 1, , 2, \ldots I$ and properly selecting $K(.)$ and $C_i^{(1)}$ (further we use a simplified notation simplified, namely C_i).

Then, (16) can be rewritten as follows:

$$\hat{\Psi}(x) \;=\; \mathrm{ROUND}\left[\frac{\sum_{i=1}^{I} i\,\hat{q}_i\, h^{-d}(n(i), i)\, K\left(\frac{||x-C_i||}{h(n(i),i)}\right)}{\sum_{i=1}^{I} \hat{q}_i\, h^{-d}(n(i), i)\, K\left(\frac{||x-C_i||}{h(n(i),i)}\right)}\right] \tag{17}$$

5 Example – A Control System for a Natural Gas Burner

In [11], [12], [1] monitoring gas burners by cameras have been discussed. As far as we know, a control system based on a camera in the loop has been proposed only in [14]. In this paper, a decision unit is based on learning an artificial neural network. Our approach is based on (17), which can be interpreted as an RBF net, but it differs from the one considered in [14] in that the structure of the decision unit is the empirical version of the optimal decisions. Thus, also its structure has some optimality advantages.

It is worth explaining why we need to control burners that are fed by a natural gas. The reason is that natural gas contains varying amounts of methane and in order to get a flame having reasonable properties we have to control the air supply rate.

Our two-stage approach is sketched in Fig. 2 (left panel), where fat lines are paths of a video stream, while thin lines are paths of transmitting signals varying in time only.

5.1 Extraction of Features from Flames

As mentioned in the Introduction, we propose the two-stage approach. The first stage consists of features extraction from a current flame. Algorithms dedicated to feature extraction and state recognition will be developed elsewhere. Here we use relatively simple features of flames. Namely, **color** i.e., the content of yellow and blue parts of the flame (see Fig. 1) and **shape of the flame** that can indicate laminar or turbulent flow of the gas and air mixture (compare images in the bottom row of Fig. 1).

At each image in Fig.1 one can notice a rotameter with its float indicating the air flow rate. The air flow rate is our input signal to the burner. Thus, the model of our system to be controlled is very simple. It consists of an input signal $u(t)$, which represents the air flow rate at time t. The flow rate is scaled from 0 (the lowest rotameter position, which corresponds to flow rate 3.5 m^3/h) to 100% (maximal air flow rate). The outputs of our system consists of two variables, namely,

– $x_1(t)$ – the percentage of the blue color in the flame at time t,
– $x_2(t)$ – the percentage of the flame height, in which the flame burns in the laminar way, i.e., it does not contain a turbulent flow.

Fig. 1 explains the meaning of $x_1(t)$ and $x_2(t)$. The flame in upper left panel contains almost no blue color ($x_1(t) = 5\%$, say) and almost all the flame shape is turbulent ($x_2(t)$ close to zero). Upper right and lower left panels show flames with more blue color and more parts with laminar shape of the flame, i.e., $x_1(t)$ and $x_2(t)$ are larger. Finally, the flame in lower right panel contains about $x_1(t) = 50$-60% of blue color and about $x_2(t) = 50$-60% of the flame length burns in the laminar way. The flame in lower right panel of Fig. 1 is considered as the reference image, i.e., the one that we would like to keep by increasing or decreasing the air flow rate independently of the methane content in a natural gas. that is supplied to the burner. One can select slightly different proportions as desired x_1^* and x_2^*, taking into account that if more air can mix with the gas before combustion, the flame burns providing a higher temperature, which results in more blue color in it. On the other hand, a smaller air supply rate leads to an incomplete reaction that appears as a light yellow flame, which is cooler.

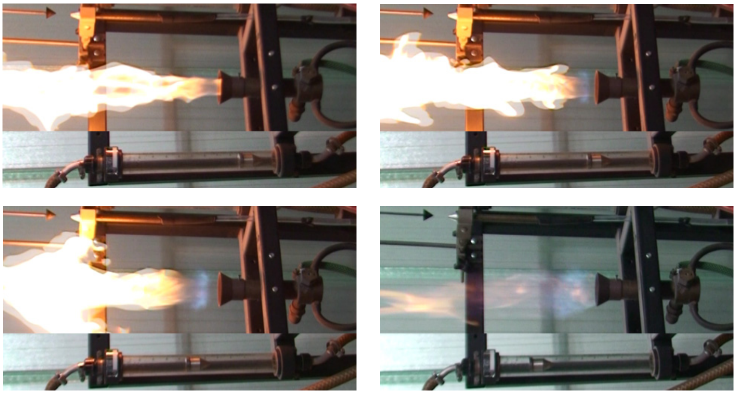

Fig. 1. Flames (rotated to save space corresponding to different air flow rates that are measured by the rotameter shown at the bottom of each figure

Analyzing the intense reactions zone of flame, the fuel undergoes pyrolysis initially, as evidenced by the presence in the area of soot and compounds C2 and CH, as manifested by a yellow color of the flame zone. Next there is an intensive zone of the oxidation reaction as the oxygen concentration increases, as evidenced by the initial absence of visible light radiation, and then the blue color of the area. This area is characterized by the highest concentrations of radicals and the highest temperature.

5.2 Control Unit

At the second stage, vectors of extracted features $x_1(t)$ and $x_2(t)$ are supplied to a control unit, where a control action, denoted as $\hat{\Psi}(x(t))$, is calculated as follows.

Table 1. Interpretation of decisions and their a priori probabilities

$$C = \left[\begin{array}{c|c} 0.1 & 0.2 \\ 0.3 & 0.4 \\ 0.5 & 0.5 \\ 0.7 & 0.6 \\ 0.9 & 0.9 \end{array}\right]$$

Dec.	Interpretation	prior q_i
dec1	increase $u(t)$ by 40%	0.1
dec2	increase $u(t)$ by 20%	0.2
dec3	keep previous $u(t)$	0.4
dec4	decrease $u(t)$ by 20%	0.2
dec5	decrease $u(t)$ by 40%	0.1

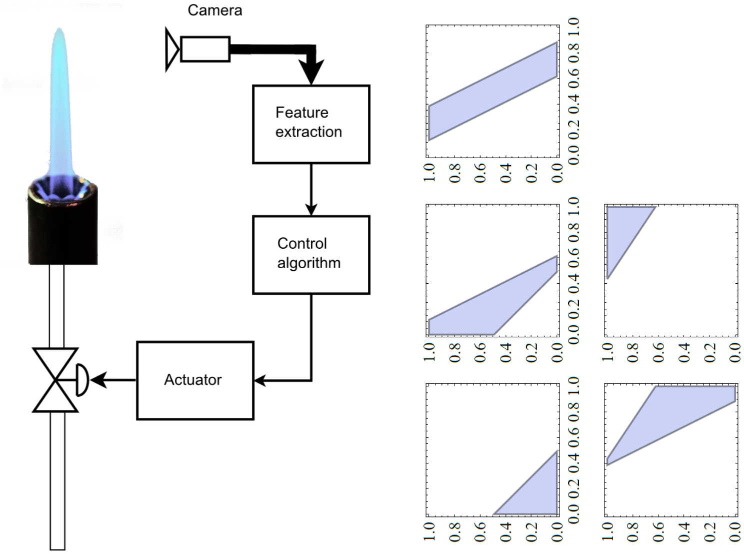

Fig. 2. Estimated optimal decision regions in Example 2 for $h = 0.005$

Consider (17) with $I = 5$ decisions, each based on $x(t) = [x_1(t),\, x_2(t)]$ and $K(.)$ being the Gaussian kernel. Matrix of centers C is selected as shown in Tab. 1 (left table).

Positions of centers correspond to the above-mentioned interpretation of $x_1(t)$, $x_2(t)$, where the row $[0.5,]\, 0.5]$ reflects the desired state x_1^* and x_2^*.

Interpretation of decisions and a priori probabilities of selecting them are shown in Tab. 1 (right table). In other words, the image based controller works as follows:

$$u(t) = u(t-1) - (\hat{\Psi}(x(t)) - 3)\, 20\%, \tag{18}$$

where $u(t-1)$ is our previous decision (it should be replaced by $u(t-\Delta)$ if the time interval between decisions is $\Delta > 0$). Finally, $u(t)$ is sent to an actuator that changes position of a valve that controls the air flow rate.

We do not present the results on selecting smoothing parameters $h(n(i), i)$, $i = 1, 2, \ldots, I$, referring the reader to [17], in which it was demonstrated that in this case $h = 0.005$ provides acceptable results. For simplicity we set all $h(n(i), i)$'s to be equal to each other. This common value will be denoted by h.

Summarizing, in Figure 2 (right panel) decision regions in the feature space are shown. We provide this version in order to stress that having decision regions one can implement the empirical decision rule as a fast controller, provided that also feature extraction is sufficiently fast.

6 Conclusions

The usefulness of linear ordering of decisions was discussed. It was shown that the minimization of the expected loss leads to a simple and interpretable decision rule. A learning algorithm is proposed that is based on the sequence of proper decisions for given stochastic states. This result is in sharp contrast with the results when 0-1 loss is minimized (see [7]). As a consequence, building 0-1 loss optimal decision rule for large number of labels is a difficult task (see, [3], [8], [9]).

Here we have provided a simple decision rule that is expected to be useful in control of industrial processes using cameras. In particular, its usefulness for control of gas burners was sketched.

Acknowledgements. This work has been supported by the National Council for Research of Polish Government under grant: 2012/07/B/ST7/01216, internal code 350914 of the Wrocław University of Technology.

References

1. Bheemul, H.C., Lu, G., Yan, Y.: Three-dimensional visualization and quantitative characterization of gaseous flames. Meas. Sci. Technol. 13, 1643–1650 (2002)
2. Bors, A.G., Pitas, I.: Median Radial Basis Function Neural Network. IEEE Trans. on Neural Networks I, 1351–1364 (1996)
3. Allwein, A., Schapire, R., Singer, Y.: Reducing Multiclass to binary: A Unifying approach for Margin classifiers. J. Machine Learning Research 1, 113–141 (2000)
4. Bishop, C.: Neural Networks for Pattern Recognition. Oxford Univ. Press (1995)
5. de Groot, M.H.: Optimal Statistical Decisions. Wiley, New Jersey (2004)
6. Devroye, L., Györfi, L.: Nonparametric Density Estimation. The L_1 View. Wiley, New York (1985)
7. Devroye, L., Györfi, L., Lugosi, G.: Probabilistic Theory of Pattern Recognition. Springer, New York (1996)
8. Dietterich, T., Bakiri, G.: Solving Multiclass Learning Problems via Error-Correcting Output Codes. J. Artificial Intelligence Research 2, 263–286 (1995)
9. Hastie, T., Tibshirani, R.: Classification by Pairwise Coupling. The Annals of Statistics 26, 451–471 (1998)
10. Karayiannis, N.B., Randolph-Gips, M.M.: On the Construction and Training of Reformulated Radial Basis Function Neural Networks. IEEE Trans. on Neural Networks 14, 835–846 (2003)

11. Kotyra, A., Wójcik, W.: Application of image flame features to assesment of combustion process. PAK 55, 117–121 (2009) (in Polish)
12. Wójcik, W., Kotyra, A.: Combustion diagnosis by image processing. Photonics Letters of Poland 1(1), 40–42 (2009)
13. Krzyżak, A., Skubalska-Rafajłowicz, E.: Combining Space-Filling Curves and Radial Basis Function Networks. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 229–234. Springer, Heidelberg (2004)
14. Lu, G., Yan, Y., Huang, Y., Reed, A.: An intelligent vision system for monitoring and control of combustion flames. Meas Control 32, 164–168 (1999)
15. Rafajłowicz, E.: RBF nets in faults localization. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 113–122. Springer, Heidelberg (2006)
16. Rafajłowicz, E., Krzyżak, A.: Pattern recognition with ordered labels. Nonlinear Analysis, Theory, Methods & Applications. Series A, Theory and Methods 17(12), 1437–1441 (2009)
17. Rafajłowicz, E.: Two stage control algorithm for industrial processes using camera. Bayesian theoretic approach, Preprint Institute of Computer Eng., control and Robotics (2013)
18. Rutkowski, L.: Adaptive Probabilistic Neural Networks for Pattern Classification in Time-Varying Environment. IEEE Trans. Neural Networks 15(4), 811–827 (2004)
19. Rutkowski, L.: Generalized Regression Neural Networks in Time-Varying Environment. IEEE Trans. Neural Networks 15(3), 576–596 (2004)
20. Skubalska-Rafajłowicz, E.: Pattern Recognition Algorithms Based on Space-Filling Curves and Orthogonal Expansions. IEEE Trans. Information Theory 47, 1915–1927 (2001)
21. Skubalska-Rafajłowicz, E.: RBF Neural Network for Probability Density Function Estimation and Detecting Changes in Multivariate Processes. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 133–141. Springer, Heidelberg (2006)
22. Xu, L., Krzyżak, A., Yuille, A.: On Radial Basis Function Nets and Kernel Regression: Statistical Consistency, Convergence Rates and Receptive Field Size. Neural Networks 4, 609–628 (1994)