Support vector machines for olfactory signals recognition

Cosimo Distantea,*, Nicola Anconab, Pietro Sicilianoa

aIstituto per la Microelettronica e Microsistemi IMM-CNR, via Arnesano, 73100 Lecce, Italy
bIstituto di Studi sui Sistemi Intelligenti per l’Automazione ISSIA-CNR, via Amendola 166/5, 70123 Bari, Italy

Received 14 June 2002; received in revised form 20 June 2002; accepted 27 June 2002

Abstract

Pattern recognition techniques have widely been used in the context of odor recognition. The recognition of mixtures and simple odors as separate clusters is an untractable problem with some of the classical supervised methods. Recently, a new paradigm has been introduced in which the detection problem can be seen as a learning from examples problem. In this paper, we investigate odor recognition in this new perspective and in particular by using a novel learning scheme known as support vector machines (SVM) which guarantees high generalization ability on the test set. We illustrate the basics of the theory of SVM and show its performance in comparison with radial basis network and the error backpropagation training method. The leave-one-out procedure has been used for all classifiers, in order to finding the near-optimal SVM parameter and both to reduce the generalization error and to avoid outliers.

Keywords: Electronic nose; Feature extraction; SVM; Radial basis function

1. Introduction

This paper focuses on the problems of detection and recognition of odors sensed by electronic nose devices. Detection, that is the problem of distinguishing a particular odor from a collection of odors, and classification, that is the problem of distinguishing several odors, are particular instances of the general problem of classification. In fact, detection and recognition can be seen as two-class and multi-class classification problems, respectively. In the context addressed in this paper, the odor classification problem is often made harder due to the irreversible behavior of the sensor array overtime such as parameter drift or noisy data. Several methods have been applied by the authors in context of parameter drift, based on adaptive resonance theory [1], or developing a SOM-based algorithm (called MSOM) that is able to re-adapt the recognition codes to the new input data distribution [2].

In the case of two-class classification problems, the main objective is to find a, in general, non-linear optimal separating surface between the two classes, starting from a collection (training set) of examples of odors belonging to the two classes. In this paper, we adopt a new category of universal feed-forward network known as support vector machines (SVM) introduced by Vapnik [3], well founded in the framework of the statistical learning theory, and appropriate for approaching classification and regression problems. The basic idea of SVM theory is closely related to regularization [4]: for a finite set of training examples, the search for the best model or approximating function has to be constrained by an appropriately small hypothesis space, that is the set of functions the machine implements. If the space is too large, functions can be found which fit the data exactly, but they will have poor generalization capabilities on new data. Vapnik’s theory formalizes these concepts and shows that the solution is found by minimizing both the error on the training set (empirical risk) and the complexity of the hypothesis space, expressed in terms of VC-dimension. In this sense, the function found by SVM is a tradeoff between closeness to the data and complexity of the solution. In particular, we show that in the case of two-class classification problems, SVM determines the optimal separating surface maximizing the margin between the two classes, that is maximizing the distance between the closest points in the training set and the separating surface. Moreover, the optimal separating surface is expressed as a linear combination of kernel functions centered on a small set of the data points called support vectors. Because, in general, support vectors are very few if compared with the cardinality of the
training set, then the representation of the optimal separating surface is sparse, in the sense that only a fraction of data points is relevant for the classification task. Moreover, the coefficients of the linear combination are determined by solving a convex quadratic programming problem with linear inequality and equality constraints. It can be shown that multilayer perception and radial basis function networks are particular cases of SVM networks, obtained by appropriate kernel functions (for a detailed description of SVM theory, see the technical reports in [5–7]). SVM has been extensively applied in computer vision, to guarantee high generalization for object recognition in cluttered environment [8]. The algorithm was able to successfully recognize an object subjected to different lighting conditions which makes the problem untractable with standard pattern recognition techniques.

SVM provides good generalization performance in the context of odor detection and classification, despite the fact that it does not incorporate problem-domain knowledge. In this paper we present the algorithm for a two-class decision problem to find the optimal setup of the machine, then we present an extension to multi-class decision problem by building several machines for each odor to be recognized. In the remainder of this paper, Section 2 presents a brief theoretical overview of SVM, Section 3 the results of the experiments, and Section 4 the conclusions.

2. Electronic nose

An electronic nose incorporates an array of chemical sensors, whose response constitutes an odor pattern. A single sensor in the array should not be highly specific in its responses but should respond to a broad range of compounds, such that different patterns are expected to be related to different odors.

A chemical sensor consists of a chemical sensitive layer and a transducer. The chemical sensitive material captures the interaction with the analyte molecules present in the environment and generates a physical change which is sensed by the transducer that converts the signal into an analog electrical output.

The transduction mechanism makes available several physical signals where electrical measurements (current, resistance, voltage, and capacitance), mass changes, heat generation, and measurements of optical changes (absorption, fluorescence, and reflectivity) have been widely used.

In an array of chemical sensors a pattern can be acquired and processed as shown in Fig. 1. Let us consider a simple odor (pure gas) or a complex one represented as a concentration vector of the \( j \)th odor class \( c_j(t) = (c_{1j}, c_{2j}, \ldots, c_{pj}) \), where \( p > 1 \) is the number of odorant components. In case of equality, we are in presence of a simple odor otherwise for \( p > 1 \) in a complex odor.

The first stage of the description of the system is the impact of the odor with the sensor surfaces. The most likely effect of the transduction process is the measurement of the electrical resistance, but in other cases could be a change in mass (for BAW sensors) or electrical potential (for Pd-gate MOSFET) [9].

The signal generated by the sensor material is then converted into an electrical signal and then conditioned. The output signal, for example, the resistance \( R_{ij}(t) \) is digitized. The converted signal is given by the vector \( y_j(t) = [y_{1j}, y_{2j}, \ldots, y_{nj}]^T \) and the array response may be preprocessed for noise and complexity reduction in order to accomplish the odor recognition task and for visualization. A typical gas sensor response is shown in Fig. 2 (ideal case), where the sensor is exposed to a certain odorant \( j \) with a certain concentration \( c_j(t) \). Usually, the rise time \( t_r \) and the decay time \( t_d \) are different. However, the output signal \( R_{ij}(t) \) is subjected to divergence from the ideal case by interfering signals.

There are several interfering inputs, the most common being changes in temperature and relative humidity of odors.
Usually, the heater of a chemoresistor is maintained at constant voltage, but in reality the operating temperature varies due to any changes in ambient temperature.

Humidity also has a strong effect on most sensors. The baseline resistance usually decreases as the humidity increases, although the exact slope depends from the operating temperature of sensors.

One of the important characteristics of an array of chemical sensors is its long-term stability. The stability is considered to be the amount of variation of the steady-state response that is represented by $R_s$ in Fig. 2 and the baseline $R_0$. The baseline is referred to the value of the referring gas injected into the vial before injecting the sample odorant for which the response changes until reaching the saturation value of $R_s$. Seasonal and environmental drifts are apparent in the behavior of some odor sensors.

Drift is a dynamical process, caused by physical changes in the sensors and the chemical background, which gives an unstable signal over the time. It is, in general, a slow change in sensitivity that occurs in time due to aging, slow morphological aspects of sensor material, poisoning, etc. Sensor poisoning is caused when the sensor is exposed to a chemical that irreversibly binds the sensing surface leading to a reduction or the total loss of sensitivity.

An example of drift is shown in Fig. 3, where a sensor is exposed to the same odorant over time. Drift could be both reversible, e.g. condensation of vapor on the sensors, and irreversible, e.g. aging that causes the pattern recognition

![Fig. 2. Characteristic response of a chemical gas sensor. The odor is presented at a certain concentration $c(t)$ and the system constituted of a sensor and the electronics converts the signal into an electrical measure, in this case change in resistance $\Delta R(t)$ is readout.](image1.png)

![Fig. 3. Example of drift in a metal oxide sensor response. Each peak represents a sensor response (measured in current changes) to repeated Pentanone exposures.](image2.png)
system to be very short-lived. Memory effect, could be considered similar to drift effect; which means that a measurement at time \( t \) is highly influenced by measurements at time \( t - k \) (for \( k \) time steps backward where the influencing odorant has been presented and measured). This leads to the fact that the same gas mixture will not give one well-defined pattern. If no drift correction of the sensor signals is made, the model will have a continuous need for recalibration. Since the training phase of the pattern recognition model should reflect the variance of the populations, many of the samples are necessary. In real application processes, the samples may be very expensive to acquire, which makes it unlikely to re-calibrate the pattern recognition model very often.

Attempts for drift reduction have been made by using a reference gas as a reference value and then correcting all subsequent readings accordingly \[10,11\]. Component correction \[12\] is a new method using one or more reference gases. This linear method is based on PCA and PLS algorithms and removes the drift direction calculated from measurements of a reference gas. Fundamental studies of the mathematical properties of the drift effects have been studied by Davide et al. \[13\]. This group has also developed successful pattern recognition models based on system identification theories \[14,15\]. These models do not require the use of a reference gas.

In order to produce consistent data for the pattern recognition process some form of preprocessing of the data from the sensor array may be necessary. Several methods have been applied as feature extraction of the sensor response, such as the difference method (\[10\]). In other words, we assume that the examples in \( S \) belong to either of the two classes. Moreover, we also assume that the classes can be separated by a hyperplane leaving all elements of a class on the same side of the hyperplane. The goal is to find the equation of the hyperplane which separates the two classes. Notice that the problem is ill-posed because the solution is not unique and then some constraint has to be imposed to the solution to make the problem well-posed.
Moreover, the constraints of the problem explicitly show that the solution is a separating hyperplane for the points of $S$. Finally, notice that the quantity $b$ enters in the constraints of the problem, but not in the objective function. The quantity $2/\|w\|$ is called the margin between the classes and the optimal separating hyperplane can be interpreted as the hyperplane which maximizes the margin (see Figs. 4 and 5). Notice that the margin is a measure of the separability of the set $S$, or, in other words, it is a measure of the complexity of the classification problem at hand.

### 3.2. Solution for linearly separable classes

The QP Problem (1) can be solved by using the standard technique of Lagrange multipliers. We need $N$ non-negative Lagrange multipliers $\lambda_i \geq 0$ each one relative to a constraint of the problem. Notice that the number of Lagrange multipliers is equal to the number of examples in the training set.

Introducing the vector $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_N)$, then the solution of the Problem (1) is equivalent to determining the saddle point of the Lagrangian function

$$L(w, b, \lambda) = \frac{1}{2}w \cdot w - \sum_{i=1}^{N}\lambda_i[y_i(w \cdot x_i + b) - 1]$$  \hspace{1cm} (6)

At the saddle point, the Lagrangian has a minimum for $w = w^*$ and $b = b^*$ and a maximum for $\lambda = \lambda^*$. Then, taking the partial derivative of $L$ with respect to $b$ we have

$$\sum_{i=1}^{N}\lambda_i y_i = 0$$  \hspace{1cm} (7)

This equation shows that at the saddle point of the Lagrangian function, corresponding to the optimal separating hyperplane $(w^*, b^*)$, the vectors $\lambda$ and $y = (y_1, y_2, \ldots, y_N)$ are orthogonal: $\lambda \cdot y = 0$. Moreover, taking the partial derivative of $L$ with respect to $w$, we have

$$w = \sum_{i=1}^{N}\lambda_i y_i x_i$$  \hspace{1cm} (8)

Substituting (8) in (6) and after some simple calculations, we have that the Lagrangian function can be written as

$$L'(\lambda) = - \frac{1}{2} \lambda \cdot D \lambda + \sum_{i=1}^{N}\lambda_i$$

where $D$ is a square matrix with size $N \times N$, $N$ being the training set size, with $D_{ij} = y_i y_j x_i \cdot x_j$ for $i, j = 1, 2, \ldots, N$. As final step, we need to maximize the Lagrangian with respect to the Lagrange multipliers and this new problem, known as the dual problem, can be formulated as follows:

**Problem 2.**

$$\max_{\lambda} - \frac{1}{2} \lambda \cdot D \lambda + \sum_{i=1}^{N}\lambda_i$$

subject to

$$\sum_{i=0}^{N}\lambda_i y_i = 0, \quad \lambda_i \geq 0, \quad \text{for } i = 1, 2, \ldots, N$$

Notice that in the dual formulation, the coefficients of the hyperplane disappear. Let $\lambda^*$ be the point where the Lagrangian takes its maximum value. Then from (8) we have

$$w^* = \sum_{i=1}^{N}\lambda_i^* y_i x_i$$  \hspace{1cm} (9)

Moreover, $b^*$ can be determined by using the Kuhn–Tucker conditions:

$$\lambda_i^*[y_i(w^* \cdot x_i + b^*) - 1] = 0, \quad \text{for } i = 1, 2, \ldots, N$$  \hspace{1cm} (10)

Let us suppose that there exists $\lambda_i^* > 0$, then from (10) follows that: $b^* = y_i - w^* \cdot x_i$. Notice that the points $x_i$, such that $\lambda_i^* > 0$, have distance $1/\|w^*\|$ from the optimal
separating hyperplane. These points are called support vectors and the problem of classifying new points \( x \) is simply solved by evaluating the following expression:

\[
sgn(w^* \cdot x + b^*)
\]  
(11)

For the case of linearly separable classes, the support vectors are the points of \( S \) closest to the optimal separating hyperplane. Moreover, they live on the margin between the two classes and they are the only points of \( S \) relevant for the classification task. In this sense, they condense all the information contained in the training set. Notice that the coefficients of the parametric representation of the optimal separating hyperplane are expressed as linear combination of the support vectors only; the points of \( S \) such that \( \tilde{z}_i^* = 0 \) do not contribute to the hyperplane representation. As a consequence, the solution of SVM for classification is sparse because, in general, only a fraction of samples live on the margin between the two classes.

Combining (11) and (9), we have that the classification of a new pattern \( x \) is solved by evaluating the decision function:

\[
f(x) = sgn \left( \sum_{i=1}^{N} \tilde{z}_i^* y_i (x_i \cdot x) + b^* \right)
\]  
(12)

that is, the class the point belongs to is obtained computing the weighted sum of dot products between the point and the support vectors.

3.3. Linearly non-separable classes

The hypothesis of linearly separable classes is too restrictive for practical applications and so we need to modify the previous results for taking into account the more general case of linearly non-separable classes. This means that there exists some point \( (x_i, y_i) \in S \) such that \( y_i(w \cdot x_i + b) < 1 \). In correspondence of this point, we can always determine a non-negative quantity \( \tilde{z}_i \) such that, \( y_i(w \cdot x_i + b) + \tilde{z}_i \geq 1 \). The introduction of this non-negative slack variable permits us to represent the constraint for a not correctly classified point in the same form of (3) and then to extend the previous analysis to the case of a linearly non-separable training set. As this aim, let us introduce \( N \) non-negative slack variables (one for each point in \( S \)) \( \xi = (\xi_1, \xi_2, \ldots, \xi_N) \) such that,

\[
y_i(w \cdot x_i + b) + \xi_i \geq 1, \quad i = 1, 2, \ldots, N
\]  
(13)

Notice that if the point \( x_i \) is correctly classified, then \( y_i(w \cdot x_i + b) \geq 1 \), the corresponding \( \xi_i = 0 \) and the constraint (13) reduces to (3). Instead, if the point \( x_i \) is not correctly classified, then the corresponding slack variable will assume a value \( \xi_i > 0 \) such that (13) is satisfied. The generalized optimal separating hyperplane for the case of linearly non-separable classes can be seen as the solution of the following problem:

**Problem 3.**

\[
\min_{w,b,\xi} \frac{1}{2} w \cdot w + C \sum_{i=1}^{N} \xi_i
\]

subject to

\[
y_i(w \cdot x_i + b) + \xi_i \geq 1, \quad i = 1, 2, \ldots, N
\]

\[
\xi_i \geq 0, \quad i = 1, 2, \ldots, N
\]

where \( C \) is a positive number. Notice that, being \( \xi_i > 0 \) for misclassified points, the term \( \sum_{i=1}^{N} \xi_i \) in the objective function of the Problem (3) is a quantity proportional to the number of misclassified points of the training set. The objective function of the Problem (3) expresses two properties of the solution in the case, of linearly non-separable classes. In fact, minimizing the first term is equivalent to maximizing the distance between the optimal separating hyperplane and the closest points in \( S \). Moreover, minimizing the second term is equivalent to minimizing the number of misclassified points. The constant \( C \), which can be regarded as a regularization parameter, controls these two terms during the training process. In fact, for small values of \( C \), the optimal separating hyperplane tends to maximize the distance of the closest point of \( S \). For large values of \( C \), the optimal separating hyperplane tends to minimize the non-correctly points of \( S \). For intermediate values of \( C \), the solution of the Problem (3) is a tradeoff between maximum margin and minimum number of misclassified points.

3.4. Solution for linearly non-separable classes

The QP Problem (3) can be solved with the technique of Lagrange multipliers. At this aim, we introduce \( N \) non-negative slack variables \( \lambda_i \) relative to the constraints \( y_i(w \cdot x_i + b) + \xi_i \geq 1 \), and \( N \) non-negative slack variables \( \mu_i \) relative to the constraints \( \xi_i \geq 0 \). If we denote with \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_N) \) and with \( \mu = (\mu_1, \mu_2, \ldots, \mu_N) \), the \( 2N \) Lagrange multipliers relative to the constraints of the Problem (3), then solving (3) is equivalent to determining the saddle point of the Lagrangian function:

\[
L = \frac{1}{2} w \cdot w + C \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \lambda_i [y_i(w \cdot x_i + b) + \xi_i - 1] \]

\[
- \sum_{i=1}^{N} \mu_i \xi_i
\]  
(14)

where \( L = L(w, b, \xi, \lambda, \mu) \). Computing the partial derivatives of \( L \) with respect to \( b \) we find (7). Moreover, computing the partial derivatives of \( L \) with respect to \( w \), we find (8). Finally, taking the partial derivative of \( L \) with respect to \( \xi_i \), we have

\[
\lambda_i + \mu_i = C, \quad i = 1, 2, \ldots, N
\]  
(15)

Substituting (8) in (14), and using the orthogonality condition (7) and the condition (15), we have that the dual of the Problem (3) can be formulated as
Problem 4.

\[
\max_k -\frac{1}{2} \lambda \cdot D\lambda + \sum_{i=1}^{N} \lambda_i
\]
subject to
\[
\sum_{i=1}^{N} \lambda_i y_i = 0, \quad 0 \leq \lambda_i \leq C, \quad i = 1, 2, \ldots, N
\]
where \( D \) is a matrix of size \( N \times N \). Notice that the only difference between separable and non-separable classes is that, in the last case, \( C \) is an upper bound of the Lagrange multipliers. In particular, for large values of \( C \), the Problem (4) reduces to the dual problem of the linearly separable case. Let \( \lambda^* \) be the optimum value of \( \lambda \). Then from (8) we have that
\[
w^* = \sum_{i=1}^{N} \lambda_i^* y_i x_i
\]
As the aim of determining \( b^* \), we use the Kuhn–Tucker conditions:
\[
(C - \lambda_i^*) \xi_i^* = 0, \quad i = 1, 2, \ldots, N
\]
\[
\lambda_i^* y_i (w^* \cdot x_i + b^*) + \xi_i^* - 1 = 0, \quad i = 1, 2, \ldots, N
\]
where \( \xi_i^* \) are the values of \( \xi_i \) at the saddle point. In particular, the value of \( b^* \) is given by: \( b^* = y_i - w^* \cdot x_i, \quad \forall i \in \mathbb{N} \). As in the case of linearly separable classes, the points \( x_i \) with \( \lambda_i^* > 0 \) are called support vectors. In particular, points \( x_i \) such that \( \lambda_i^* = C \) are called errors. Finally, as in the case of linearly separable classes, the classification of a new data \( x \) involves the evaluation of the decision function:
\[
f(x) = \text{sgn} \left( \sum_{i=1}^{N} \lambda_i^* y_i (x_i \cdot x) + b^* \right)
\]
where the solution is expressed evaluating the dot product between the data and some elements (support vectors) of the training set \( S \).

3.5. Non-linear separating surfaces

So far we have limited our analysis to the case of linear separating surfaces and we have showed that the optimal separating hyperplane can always be expressed as weighted sum of dot products between the new data point and the support vectors. The extension of the theory to more complex separating surfaces is simple and it is done by mapping the input patterns \( x \) in a higher dimensional space, called feature space, and looking for an optimal separating hyperplane in this new space. In a more formal way, the input pattern \( x \) is mapped in a (possibly infinite) feature vector:
\[
(a_1 \phi_1(x), a_2 \phi_2(x), \ldots, a_n \phi_n(x), \ldots)
\]
where \( a_n \in \mathbb{R}^\infty \) are real numbers and \( \phi_i \) are real functions. After this mapping, it is possible to apply SVM by using \( \phi(x) \) as input vectors, and then, in this new space, the solution of SVM has the form
\[
f(x) = \text{sgn} \left( \sum_{i=1}^{N} \lambda_i^* y_i (\phi(x_i) \cdot \phi(x) + b^*) \right)
\]
An important property of SVM is that the only relevant quantities to be computed for establishing the class of a new pattern are scalar products of the form \( \phi(x) \cdot \phi(y) \). To this aim, it is convenient to introduce the so-called kernel function \( K \) defined in the following way:
\[
K(x,y) = \phi(x) \cdot \phi(y) = \sum_{n=1}^{\infty} a_n^2 \cdot \phi_n(x) \cdot \phi_n(y)
\]
By using this quantity the solution of SVM becomes
\[
f(x) = \text{sgn} \left( \sum_{i=1}^{N} \lambda_i^* y_i K(x_i, x) + b^* \right)
\]
Notice that, in this case, the separating surface is a non-linear function represented as a linear combination of kernel functions, each one centered on a support vector. The kernel function we used is:
\[
K(x,y) = (1 + x \cdot y)^d
\]
where \( d \) is an integer number. In general, admissible kernel functions are positive definite functions verifying the Mercer’s theorem [17].

4. Experimental setup

SVM provides good generalization performance in the context of odor detection and classification, despite the fact that it does not incorporate problem-domain knowledge.

The active layers of the sensor array consist of pure and doped SnO_2 thin films prepared by means of sol–gel technology. Pd, Pt, Os, and Ni were chosen as doping elements starting from different precursors of the preparation of the modified films. The films, whose thickness was about 100 nm, were deposited on alumina substrates supplied with interdigitated electrodes and platinum heater, by the spin coating technique at 3000 rpm, dried at 80 °C and heat treated in air at 600 °C. After deposition, the sensors were mounted onto a TO8 socket and inserted in the test chamber. The conditioning of sensors was performed by using dry air (flow 100 sccm) as reference gas for acquiring the baseline. The acquired training set was constituted of several odorants like pentanone, hexanal, water, acetone and three mixture as a combination, some of them are shown in Table 1.

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Three mixtures considered in the data set</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>Mixture</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>M1</td>
</tr>
<tr>
<td>M2</td>
</tr>
<tr>
<td>M3</td>
</tr>
</tbody>
</table>
5. Experiment results

The dataset is shown in Figs. 6 and 7, in the first two principal component space. Notice that principal component is only used to visualize the dataset, since to the classifier is given the raw signal by using the Relative method \( \frac{I}{I_0} \), where \( I \) represents the steady-state current measurement and \( I_0 \) the relative baseline value) as feature extraction.

![Fig. 6. Dataset projected on the first two principal components and relative variance captured.](image1)

![Fig. 7. Each machine is associated to an odor in the data set and trained against the others. The near-optimal regularization parameter C for each machine is found by finding the minimum leave-one-out error.](image2)
In a first analysis, we used a SVM having a second degree polynomial kernel function. Moreover, the optimal regularization parameter \( C \) of the SVM was found out experimentally by minimizing the leave-one-out error over the training set, which provides an estimate of the generalization performances of the final classifier. Each machine has been trained to solve a two-class problem, in order to find experimentally the regularization parameter \( C \) seen above. The multi-class problem for a two-class trained machine (SVM) is carried out with leave-one-out procedure and Eq. (22) gives an evaluation of the winning machine for \( (SVM) \) is carried out with leave-one-out procedure and Eq. (22) gives an evaluation of the winning machine for each test data point. It will be shown the superiority of the SVM-based classifier for the complex dataset and the leave-one-out training process suitable for small training sets and for avoiding outliers in the learning of the functional input-output mapping. As a comparison, two further feed-forward-neural networks have been used with error back propagation and radial basis function. Comparing the prediction error, we found that back propagation method gave poorer results with more than 40% error, followed by the RBF network with 15%, then SVM with 4.5%. The Table 2 is shown the confusion matrix of the SVM classifier. Rows indicate true values and columns, those respectively predicted. As can be noted in the table, SVM makes some mistakes in predicting mixtures M1, M2, and M3 against each other.

As a benchmark with other classifier systems, two feed-forward-neural networks have been considered: error back-propagation algorithm and radial basis function [18] with the same SVM conditions for training and testing data sets.

The RBF network used, creates neurons one at a time. At each iteration, the input vector which will result in lowering the network sum-squared error is used to create a new radial basis neuron. The error of the new network is checked, and if low enough, the learning phase is finished. Otherwise the next neuron is added. This procedure is repeated until the error goal is met (0.001), or the maximum number of neurons is reached (i.e. the number of training vectors). A spread of 0.8 is used for radial basis functions in order to ensure that more than one neuron can respond to overlapping regions of the input space. RBF network structure at the end of the training phase was \( 3 \times 157 \times 6 \) and the error reported for the testing phase was 49% as shown in Table 3.

6. Conclusions

In this paper, we have adopted a novel classifier for odor recognition with complex patterns. The objective of the classifier is to find optimal hyperplanes for separating clusters in the non-linearly separable context. The leave-one-out procedure has been used, in order to approach as much as possible the generalization error with the given dataset, and also avoiding outliers. The method has also been compared with other two known classifiers such as RBF and a backpropagation training method, that provided lower performant results. In this paper, a complex dataset is used in the case of the so-called standard situation (the probability distribution of the dataset does not change with time), so then future works will be addressed on the use of SVM extended to support problems where the signal is subjected to drift.

References


---

Table 2

Confusion matrix for SVM classification results with leave-one-out procedure, true vs. predicted (rows vs. columns)

<table>
<thead>
<tr>
<th>Water</th>
<th>Acetone</th>
<th>M1</th>
<th>Hexanal</th>
<th>M2</th>
<th>M3</th>
<th>Pentanone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water</td>
<td>28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Acetone</td>
<td>0</td>
<td>28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>M1</td>
<td>0</td>
<td>0</td>
<td>33</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hexanal</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>34</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>M2</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>32</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>M3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>Pentanone</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>24</td>
</tr>
</tbody>
</table>

Table 3

Confusion matrix for RBF classification results with leave-one-out procedure, true vs. predicted (rows vs. columns)

<table>
<thead>
<tr>
<th>Water</th>
<th>Acetone</th>
<th>M1</th>
<th>Hexanal</th>
<th>M2</th>
<th>M3</th>
<th>Pentanone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water</td>
<td>19</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Acetone</td>
<td>0</td>
<td>26</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>M1</td>
<td>0</td>
<td>0</td>
<td>27</td>
<td>1</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>Hexanal</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>33</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>M2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>31</td>
<td>0</td>
</tr>
<tr>
<td>M3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>34</td>
<td>0</td>
</tr>
<tr>
<td>Pentanone</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>23</td>
</tr>
</tbody>
</table>
Recognition and Image Processing, Atlantic City, USA, 27 February to 3 March 2000.


