NONLINEAR FILTERS BASED ON SUPPORT VECTOR MACHINES

David A. Márquez, José L. Paredes, Winston García-Gabín
Electrical Engineering Department, Universidad de Los Andes. Mérida 5101, Venezuela
{damarquezg, paredesj, winstong}@ula.ve

ABSTRACT
In this work, we take advantage of the classification capability of SVM in a binary filtering process, replacing the Boolean function that characterizes the Boolean filter by a Support Vector Machine, defines thus a new family of nonlinear filters. The success of these filters is based on two intrinsic properties: edge preservation and efficient noise attenuation, being robust against impulsive noise.

2. DEFINITIONS AND PRELIMINARIES

2.1. Threshold decomposition
Threshold decomposition is a powerful tool used for the analysis and implementation of Boolean filters including in this family. It provides a simple way to obtain good classification results with reduced a priori knowledge of the problem, being the most used application, the use of SVM as binary classifier where the observation vector is categorized in one out of two possible classes.

In this work, we take advantage of the classification capability of SVM in a binary filtering process, replacing the Boolean function that characterizes the Boolean filter by a Support Vector Machine, defines thus a new family of nonlinear filters. The success of these filters is based on two intrinsic properties: edge preservation and efficient noise attenuation, being robust against impulsive noise.

2.2. Boolean Filters
Let \( X(n) \) be the observation vector at \( n \)-th window position, a Boolean filter, denoted by \( TBF_f(X(n)) \), is defined as [1]:

\[
TBF_f(X(n)) = \frac{1}{2} \sum_{m=-M}^{M} f[T^n(X(n))]
\]

A new family of non-linear filters based on SVM is proposed in this paper. The potential of these filters is showed in two image applications, image denoising and edge detection. In the latter, a new structure to detect edges using a non-traditional approach based on the proposed filters is developed. Different from the traditional methods where the derivative is approximated or complex mathematical methods are used to detect edges, the main idea, in our approach, is to train the SVM in the binary domain created by a threshold operator, so that it can recognize the presence of edges in the image of interest. The results show that our approach is better than traditional edge detectors in particular when the images are corrupted by impulsive noise and yields competitive results on clear images.
where \( X(n) = [X_1(n), \ldots, X_N(n)] \), with \( X_i(n) = X(n - (N-1)/2 + i - 1) \), \( f : \{-1,1\}^N \rightarrow \{-1,1\} \), and \( T^m(\tilde{x}) \) is the threshold operator defined in Eq. (1). Note that the filtering operations reduces to decomposing the observation vectors in its binary representation, filtering each binary vector using the binary filter \( f(\cdot) \), adding up all the binary outputs to form the filter output. This family of filters is completely specified in the binary domain through the truth table of \( f(\cdot) \) or its corresponding minimum sum of products.

Boolean filters include stack filters as a special case [2], [8]. A stack filter is a particular type of Boolean filter defined by Eq. (2) where \( f(\cdot) \) is a positive Boolean function that satisfies the stacking constrain [2], [8].

2.3. Support Vector Machine (SVM)

SVM principles were developed by Vapnik and presented in several works as in [5], [9]. Consider a binary classification problem, where a collection of vectors \( (X \in R^d) \) are available. Furthermore, consider that the set of vectors are related to two different classes, \( y_1 \) and \( y_2 \), and it is desired to find an optimal hyperplane to divide these classes. The optimal decision boundary will be the one that maximizes the distance from the hyperplane to the training data. In the two dimensional case, the hyperplane will be a line, while in a multidimensional space, the hyperplane will be so that, \( (W \cdot X) + b = 0 \), where \( W \in R^d \) and \( b \in R \) are obtained as the solution of the optimization problem [9]:

\[
\begin{align*}
\min_{W, b} \frac{1}{2} ||W||^2 \\
\text{subject to: } y_i \left( \langle W \cdot X \rangle + b \right) \geq 1, \quad i = 1, \ldots, l
\end{align*}
\]  (3)

The decision hyperplane can be written as:

\[
\begin{align*}
f(X) = \text{sgn} \left( \sum_{i=1}^{N} a_i y_i \langle X_i \cdot X \rangle + b \right)
\end{align*}
\]  (4)

where \( y_i \in \{+1, -1\} \) defines the class where \( X_i \) belongs. Thus, a set of training data is needed \( \{ (X_i, y_i) \}_{i=1}^{l} \) in order to find the classification boundary. In Eq. (4), \( a_i \)’s are Lagrange multiplicators obtained as part of the solution of the constrained optimization problem and \( l \) represents the number of training samples used to define the decision frontier vectors. The vectors \( X_i \), for \( a_i \neq 0 \) are known as support vectors since the separation region are defined by those vectors [4].

When the training data are not linearly separable, this scheme cannot be used directly. In order to solve this problem, SVM turns the entry observation vector into a characteristic space of a higher dimension, solving the optimal problem in such space, and returning to the original space converting the optimal hyperplane in a non-linear decision frontier [4]. The non-linear expression to the classification function is given by:

\[
\begin{align*}
f(X) = \text{sgn} \left( \sum_{i=1}^{m} a_i y_i K(X_i, X) + b \right)
\end{align*}
\]  (5)

Where \( K(\cdot, \cdot) \) is a kernel function that performs the non-linear transformation on the observation vectors.

In practice, an optimal separating hyperplane may not exist, in this case the optimization problem is solved by inserting non-negative slack variables \( \xi \), reducing the optimization problem to:

\[
\begin{align*}
\min_{W, b, \xi} \frac{1}{2} ||W||^2 + C \sum_{i=1}^{m} \xi_i \\
\text{subject to: } y_i \left( \langle W \cdot X \rangle + b \right) \geq 1 - \xi_i, \quad i = 1, \ldots, l
\end{align*}
\]  (6)

where \( C \) is a penalty term that makes more or less important the misclassification error in the minimization process and, therefore, it is a tuning parameter [4].

3. SUPPORT VECTOR MACHINE FILTERS (SVMF)

For a boolean filter defined by Eq. (2), it is possible to replace the function \( f(\cdot) \) that characterizes the Boolean filter by a decision function corresponding to a SVM, defining thus a new family of non-linear filters.

Let \( X = \{X_1, X_2, \ldots, X_N\} \) be the observation vector to be filtered, furthermore, let \( x^m = \left[ x_1^m, x_2^m, \ldots, x_N^m \right] \) be its correspondent threshold decomposition at threshold level \( m \).

The output of the Support Vector Machine filter, denoted by SVMF(X), is defined as:

\[
\begin{align*}
\text{SVMF}(X_1, X_2, \ldots, X_N) = \frac{1}{2} \sum_{m=0}^{N} \text{SVM}_f(x_1^m, x_2^m, \ldots, x_N^m)
\end{align*}
\]  (7)

where \( \text{SVM}_f(\cdot) : \{-1,1\}^N \rightarrow \{-1,1\} \), is a decision function corresponding to a SVM as in Eq. (4) and \( x^m = T^m(X) \), is the threshold decomposition of input vector \( X \).

Substituting \( \text{SVM}_f(\cdot) \) for the decision structure from Eq. (5), Eq. (7) reduces to:

\[
\begin{align*}
\text{SVMF}(X) = \frac{1}{2} \sum_{m=0}^{N} \left[ \text{sgn} \left( \sum_{i=1}^{l} y_i a_i K(x^m_i, x_i) + b \right) \right]
\end{align*}
\]  (8)

At first look, it seems that the computational cost of Eq. (8) is expensive, this, however, can be notably reduced if it is noticed that for any \( m \in \{ -\infty, X(0) \} \) or \( m \in \{ X(1) - X(0), X(2) \}, i = 2, \ldots, N \) or \( m \in \{ X(N) + \infty \} \), threshold decomposition outputs the same binary vectors. Therefore, there are at least \( N+1 \), different binary vectors \( x^m \). Thus, after some simplifications, Eq. (8) reduces to:

\[
\begin{align*}
\text{SVMF} = \frac{X(N) + X(0)}{2} + \frac{1}{2} \sum_{m=0}^{N} \left( X(1) - X(0) \right) \text{sgn} \left( \sum_{i=1}^{l} y_i a_i K(x^m_i, x_i) + b \right)
\end{align*}
\]  (9)

where \( X(0) \) is the \( i \)-th smallest sample of the set \( \{X_1, X_2, \ldots, X_N\} \), with \( X(1) \leq X(2) \leq \ldots \leq X(N) \). The filter representation in Eq. (9) provides us with an interesting interpretation of the SVM filter. The filter output is computed by the sum of the midrange of the signed samples \( \{X(1) + X(0)/2 \} \) and a linear combination of the differences between successive order statistics \( \{X(i) - X(i-1)\} \), multiplied by a factor \( \pm(1/2) \) whose signed depends on the training samples. This is another way to see the proposed SVM filter and, as expected, output the same results as the one obtained by Eq. (8).

4. SVM FILTER DESIGN BASED ON DETAIL STRUCTURE

Several 3x3 masks were designed generalizing some particular cases that may appear in a filtering process. Thus, a 9-component
vector are formed and used to train the SVM that, in turns, defines the filtering characteristic function.

Figure 1 depicts the designed masks. These masks were created trying to get a good prediction model, expecting the SVM to generalize to other possible cases that may appear during the filtering process. On the upper part of each mask is the assigned label that represents the desired output for that particular mask. For instance, the training vector for the first mask of Fig. 1, is \([X; y] = [1,1,-1,1,1,-1,1,1,-1;1]\). The assigned label to this particular mask is “+1” which indicates that the white zone is generalized. Thus a vertical line is preserved during the filtering operation.

5. APPLICATIONS

5.1. Impulsive noise removing using SVM filter

As a first application, the proposed filter is used in an image denoising task, more precisely to mitigate the impulsive noise in images. SVM filters are trained using masks like the ones show in Fig. 1, for the linear kernel of the type, \(K (X, Y) = \langle X - Y \rangle\) and a radial base function, \(K (X, Y) = \exp (-\frac{|X - Y|}{2\sigma^2})\).

The performance of the proposed filter is compared to the performance obtained using Center Weighted Median (CWM) filters [3]. The best CWM filter with minimum Mean Square Error (MSE) and Mean Absolute Error (MAE) is chosen and used in the comparison. Thus the performance of the SVM filter is compared to the performance of the best CWM filter.

Figure 2 shows the performance of the SVM filter with RBF kernel, for this case the parameter \(C\) changed from \(C = 10^{-2}\) to \(C = 10^2\) obtaining the best result for \(C = 1\) and \(\sigma = 10\). Figure 2 also shows a zoom-in of the image part enclosed by a rectangle. As can be seen, the SVM based filter has a better performance, eliminating impulsive noise efficiently, while preserving details and features present on the original image.

Table 1. Mean Square Error and Mean Absolute Error.

<table>
<thead>
<tr>
<th></th>
<th>MSE SVM Filter</th>
<th>MSE CWM Filter</th>
<th>MAE SVM Filter</th>
<th>MAE CWM Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cameraman</td>
<td>65.69</td>
<td>75.09</td>
<td>1.74</td>
<td>2.31</td>
</tr>
<tr>
<td>(linear kernel)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cameraman</td>
<td>60.65</td>
<td>75.09</td>
<td>1.61</td>
<td>2.31</td>
</tr>
<tr>
<td>(RBF kernel)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Circuit</td>
<td>48.49</td>
<td>49.13</td>
<td>1.25</td>
<td>1.27</td>
</tr>
<tr>
<td>(RBF kernel)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lena</td>
<td>43.05</td>
<td>46.19</td>
<td>2.18</td>
<td>2.71</td>
</tr>
<tr>
<td>(RBF kernel)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.2. Image edge detection based on SVMF

As a second application, an approach for edge detection is presented using a different point of view from the traditional [10]. In our case, we do not try to approximate the derivative or use other mathematical methods to detect edges in an image. The main idea, in this case, is to train the SVM filter to recognize the presence of edges in an image. This application derives from the design based on edge and structures to be preserved.

In order to get an appropriate generalization, a new set of 3x3 masks are designed. Figure 3 depicts some of the designed masks that are used to train the SVM corresponding to the decision structure filter. The masks are designed such that the edge information of the image is captured by the filter while the impulsive noise is removed. Three groups of training vectors were created to detect: vertical, horizontal and diagonal edges and they are used to train three SVM. The kernel functions used in the SVM is the RBF kernel.

The designed masks allow that edge detection using the SVM filter can be generalized to any kind of image either non-noisy or noisy ones.

For a SVM filter denoted by Eq. (8), it is possible to define a function that permits to detect edges in images. SVM filter for edge detection output is defined as:
\[ SVM_{\text{EDGE}} = \begin{cases} +1 & \text{if } \sqrt{(E_h)^2 + (E_v)^2 + (E_d)^2} \geq Th \\ 0 & \text{otherwise} \end{cases} \]  

(10)

where \(E_h\), \(E_v\) and \(E_d\) correspond to the output of the three SVM filters, as in Eq. (8), trained using masks like the ones show in Fig. 3, that detect horizontal, vertical and diagonal edges, respectively. The value of \(Th\) is a tunable parameter that can be adjusted as a tradeoff between the amount of edges to be detected and the noise immunity. High threshold values yield robustness to impulsive noise but lose some of the real edges, whereas low values result in many false edges induced by the impulse noise. The outputs of the three SVM are combined to find the total amount to which any edge exist, this value is the compared to the threshold \(Th\) to determine the existence of an edge.

The performance of the proposed edge detection approach is compared to those yielded by traditional edge detection methods, among them: Sobel, Canny, Prewitt and Roberts, which have been widely used in digital image processing [11].

Figure 4 shows SVM filter performance in edge detection in noise images. Edge detection using SVM filter is superior to the rest of the methods. The image obtained in Fig. 3 (b) reflects the generalization capabilities of the proposed method since SVM filters not only eliminate impulsive noise effectively, but also many details (edges) from the original image are suitably detected. A second example that shows the performance of the proposed filter in edge detection in noise images is shown in Fig. 5, as before, it is easy to see the superior performance of the edge detection method using SVM filter compared to those yielded by traditional edge detection method. Note that most of the edges are preserved while the noise is removed.

6. CONCLUSIONS

In this work, a new family of non-linear filters based on SVM is presented. The proposed filter is based on the general concept of binary filters where the characteristic function of the Boolean filter is replaced by a SVM. The obtained results for the new filtering approach show a better performance than traditional methods either in the noise elimination or the edge detection, behaving as a competitive filter for digital image processing.

7. REFERENCES


