# Detection and Classification of Underwater Transients with Data Driven Methods Based on Time-Frequency Distributions and Non-Parametric Classifiers

Paulo M. Oliveira (*), Victor Lobo (*), Victor Barroso (**) and Fernando Moura-Pires (***)

* ESCOLA NAVAL
Alfeite, 2800 Almada, Portugal
pmonica@mail.telepac.pt
vsl@di.fct.ul.pt

**ISR/IST
Torre Norte, Piso 7 Av. Rovisco Pais
1049-001, Lisboa, Portugal
vab@isr.ist.utl.pt

*** Universidade de Évora
Departamento de Informática
Évora, Portugal

***Abstract*-Due to the complexity of underwater transients and background interference, model based approaches to transient detection/classification are often not practical. This has motivated an interest for data-driven, model-free methods. One such method was presented in [2] and modified in [1], where it was applied to the detection of underwater transients. In this article, we will extend that approach, to allow its use in the more demanding environment of a brown water environment, where background noise is constituted by a multitude of different interferences, non-white, and highly non-stationary. Also, the assumption of linear separability amongst the transients and the background noise in the time-frequency or related domains will be discarded, leading to the use of an additional classifier stage. A technique to minimize the number of prototypes on this classifier will be presented. The developed methods are used to detect and classify real underwater transients, recorded off the Portuguese coast. Estimation of the overall error rate of the method is obtained using cross-validation with the available data set, showing that these methods can effectively be used in real environment situations.**

## I. INTRODUCTION

In [1], a time-frequency based, data driven method was used to classify underwater acoustic transients. The method was based in a proposal made in [2], where use is made of the fact that any bilinear time and frequency shift covariant time-frequency distribution $\rho_S(t,f)$ (this class of distributions is usually referred to as Cohen's class) can be interpreted as the result of "smoothing" the Wigner-Ville distribution (WD) with a 2D kernel function $\Psi(t,f)$ [3]. That is,

$$\rho_S(t,f) = \iint \Psi(\mu - t, \nu - f)\, WD_S(\mu,\nu)\, d\mu\, d\nu, \qquad (1)$$

where

$$\Psi(t,f) = \iint \phi(\theta,\tau)\, e^{j2\pi(\theta t + \tau f)}\, d\theta\, d\tau. \qquad (2)$$

Each choice of the kernel $\phi(\theta,\tau)$ will originate a different time-frequency distribution, to which we will refer as a *generalized time-frequency distribution.*

The link between generalized time-frequency distributions and optimal detectors comes from the fact that the Wigner-Ville distribution obeys Moyal's formula [4]:

$$\iint WD_p(t,f) WD_S(t,f)\, dt\, df = \left| \int p(t) s^*(t) dt \right|^2 \qquad (3)$$

This means that we can implement the optimal quadratic detector by computing the inner product of the WD of the received signal with the WD of the prototype signal to be detected. Since the WD is time/frequency shift covariant, a time/frequency shifted $WD_p(t,f)$ is the WD of the correspondingly time/frequency shifted prototype. This means that, under the assumption of stationary white noise, the double convolution of the WD of a signal with a time-frequency reversed WD of the prototype generates a set of optimal quadratic detectors covering all points of the delay-doppler space. Considering (1), we conclude that any generalized time-frequency distribution can be interpreted as a set of optimal quadratic detectors tailored to the particular signal whose WD is the function $\Psi(t,f)$ smoothing $WD_S(t,f)$, the WD of the received signal [2]. Since, by construction, all delay-doppler space is searched, what we have, in fact, is a scheme whose detection capability is delay-doppler independent. This method has been applied to real signals, collected in the controlled environment of an acoustic tank, with good results [1].

## II. COMPUTATIONAL COST

The computational cost to obtain the set of quadratic detectors represented by $\rho_S(t,f)$ is very high. The straightforward approach to the method would imply computing the WD of the received signal, and then its smoothing with the WD of the prototype. The number of multiplications involved in such a direct implementation of (1) is of the order $N^4$. One possible approach to decrease the computational cost of the method relies on the possibility of computing directly the generalized time-frequency distribution $\rho_S(t,f)$, avoiding the intermediate steps of

This work was partially supported by the FCT Programa Operacional Sociedade de Informação (POSI) in the frame of QCA III, under contract POSI/32708/CPS/2000

evaluating the WD and applying the smoothing kernel for the desired distribution. Some of these generalized distributions can be computed directly from the signal with reasonably low computational costs. The Margenau-Hill distribution $MH_S(t,f)$, for example, can be directly computed with a total cost of approximately $2N^2$+NlogN, which is considerably faster than obtaining it by using (1) with the appropriate smoothing kernel.

However, in our application, we do not have the freedom to choose a particular distribution. To obtain the set of optimal quadratic detectors for a given transient, $\Psi(t,f)$ in (1) must be the WD of the transient to be detected. According to (2), this uniquely determines the kernel function $\phi(\theta,\tau)$ to be used. Therefore, we will not be able to decrease the computational effort by conveniently choosing the distribution to be used. We can, however, approximate the generalized distribution needed by a linear combination of fastly computable distributions. If the number of basis distributions necessary for an adequate approximation is small enough, we may gain some benefits in the overall computational cost of the method.

Such a method was analyzed in [5], where generalized distributions were implemented as linear combinations of spectrograms. Since spectrograms are relatively fast to compute, considerable computational gains can be obtained. This depends, of course, on the number of spectrograms that one must compute to build an acceptable approximation to the desired generalized distribution.

All bilinear time-frequency distributions can be written as

$$\rho_S(t,f) = \iiint s(u+\frac{\tau}{2})s^*(u-\frac{\tau}{2})\phi(\theta,\tau)e^{-j2\pi(\theta t-\tau f+\theta u)}\,du\,d\theta\,d\tau,$$

from where we easily obtain

$$\rho_S(t,f) = \iint s(u+\frac{\tau}{2})s^*(u-\frac{\tau}{2})\,\Lambda(u-t,\tau)e^{-j2\pi f\tau}\,du\,d\tau, \qquad (4)$$

$\Lambda(t,\tau)$ being the inverse Fourier Transform of the kernel function w.r.t $\theta$,

$$\Lambda(t,\tau) = \int \phi(\theta,\tau)e^{j2\pi\theta t}\,d\theta. \qquad (5)$$

For real valued time-frequency distributions,

$$\phi(\theta,\tau) = \phi^*(\theta,-\tau)$$

and then

$$\Lambda(t,\tau) = \Lambda^*(t,-\tau).$$

Hence, under very mild conditions, $\Lambda(t,\tau)$ can be decomposed into an orthonormal basis of eigenfunctions $u_k$, corresponding to real eigenvalues $\lambda_k$, as follows (see [5] for details):

$$\Lambda(t,\tau) = \sum_k \lambda_k u_k(t+\frac{\tau}{2})u^*_k(t-\frac{\tau}{2}). \qquad (6)$$

Substituting (6) in (4), we conclude that

$$\rho_S(t,f) = \sum_k \lambda_k \left|\int s(\xi)u^*(\xi-t)e^{-j2\pi f\xi}d\xi\right|^2.$$

This means that real valued generalized distributions can be decomposed into a sum of spectrograms.

This decomposition can really decrease the computational cost, depending on how fast the eigenvalues $\lambda_k$ decrease and, therefore, on how many spectrograms are needed to maintain a reasonable degree of approximation. This also depends on the particular generalized time-frequency distribution and, consequently, on the transients to be detected.

In this article, we will be classifying two different transients, recorded on shallow waters in a brown water environment, where background noise is constituted by a multitude of interfering man-made noises, non-white, and highly non-stationary. One of the transients ($\mathbf{tr_1}$) is the sound produced by hitting a metallic tube with a metallic hammer, and the other one ($\mathbf{tr_2}$) was produced with the same tube being hit with a rubber hammer. Using a simple perceptron, trained with Rosemblat's algorithm (see e.g. [6], for details on the perceptron concept, design and training), we obtained the $\Lambda(t,\tau)$ corresponding to one of the transients (hit with the metallic hammer). Its first 60 eigenvalues can be seen in Fig. 1.

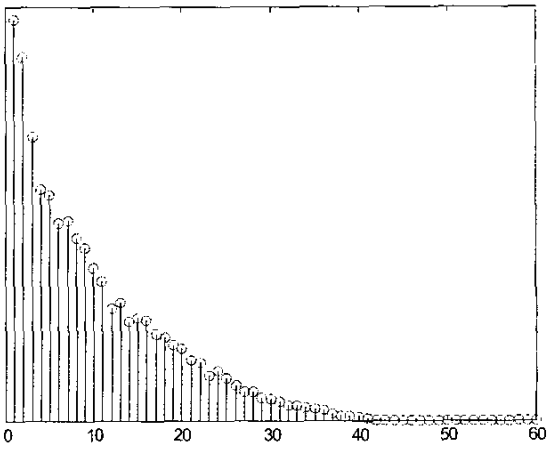


**Fig. 1 First 60 eigenvalues of $\mathbf{tr_1}$.**

As can be seen, there is not a small number of dominant eigenvalues. This means that, to approximate the generalized time frequency distribution corresponding to this transient, we would need a large number of periodograms, and no computational savings could be obtained. The same can be said for $tr_2$.

We thus conclude that, in the particular case of the transients considered in this article, no computational savings will be obtained by this technique of decomposition in periodograms.

As an alternative, we can decrease the computational effort of the procedure by working in the ambiguity plane. The ambiguity plane is the double inverse Fourier Transform of the time-frequency plane. This means that, in this domain, the convolution in (1) can be implemented with a cost of $4N^2\log N+N^2$ multiplications, corresponding to the direct and inverse 2D FFT's, plus $N^2$ multiplications.

The full procedure to implement the set of quadratic detectors in (1) thus becomes:

a. Compute the Ambiguity function $A(\theta,\tau)$ of the received signal (approximately $N^2+N^2\log N$ multiplications):

$$A(\theta,\tau) = \int s(t+\frac{\tau}{2})s^*(t-\frac{\tau}{2})e^{j2\pi\theta t}dt;$$

b. Multiply $A(\theta,\tau)$ by the (pre-computed) ambiguity domain kernel $\phi(\theta,\tau)$ corresponding to the particular transient to de detected (approximately $N^2$ multiplications);

c. Take the 2D IFFT to transform back from the ambiguity domain to the time-frequency domain (approximately $2N^2\log N$ multiplications);

$$\rho_S(t,f) = \iint A(\theta,\tau)e^{j2\pi(\theta t+\tau f)}d\theta d\tau$$

In general, and unless the particular transients to be detected admit a decomposition in a small number of periodograms, this is the best approach from a computational viewpoint.

## III. NEAREST NEIGHBOR CLASSIFICATION

In cases where there is very little information about the probability distribution of the data, and when that data is highly irregular, non-parametric classification techniques tend to outperform parametric ones [7]. The most popular technique for non-parametric classification is probably the nearest neighbor rule ([8], [9]) and it's derivates. To use a nearest neighbor classifier, one must have a set of data patterns for which the class, or label, is known. This set is called the training set, and in a nearest neighbor classifier the entire set is stored without any processing. These stored patterns are sometimes called prototypes. When the system is called upon to classify a new data pattern, the distance to all stored patterns is calculated, and the new pattern is assigned the same label as it's nearest neighbor.

With the ever growing memory and processing power of computers, nearest neighbor classifier have become more atracticve, and a great deal of research on this area has been made [10], [11]. The basic concept of nearest neighbor classifiers has been used in a variety of different areas with many different names, such as lazy learning or memory-based systems [12].

One of the main advantages of the nearest neighbor rule is it's theoretical soundness and the fact that asymptotically, as the number of training patterns grows, the expected error rate will converge to less the twice the optimum bayes error ([9], [13]). Even when using finite training sets, it has been proved that the expected error is very close to that value [14]. However there are a few major drawbacks in using nearest neighbor classifiers:

a) Memory requirements are very large, since the entire training set must be stored;
b) Processing requirements during classification are very demanding, since the similarity between the new data pattern and all stored patterns must be computed;
c) The classifier is quite sensitive to outliers.

Many improvements have been proposed to address these shortcomings, and they can broadly be summarized into data editing techniques that aim at eliminating outliers, and data condensing techniques, aiming to decrease the number of stored patterns. In this paper, we will present a data condensing technique, called Q-set minimization [15], which can be used to select a reduced set of patterns to form a simple, yet efficient, nearest neighbor classifier.

## IV. Q-SET MINIMIZATION

Q-set minimization can be done using what are called positive-only functions, or generalized functions. In this paper we will use the simpler positive-only functions. The positive-only Q-set algorithm may be summarized as follows:

### Algorithm 1 - Computing Positive-only *Q*-sets

**Let**

$X_{train}$ be the set of training patterns $\boldsymbol{x}$
$\boldsymbol{P}$ be the set of candidate prototypes $\boldsymbol{p}$
$\boldsymbol{Q}$ be a vector with the $\boldsymbol{Q}$-sets of each pattern, initialized to $\varnothing$
$\boldsymbol{P_{sel}}$ be the set of indexes of the selected prototypes $|Y|$ be the cardinality of some set $Y$

**Do**

1 For i=1 to $|\boldsymbol{X_{train}}|$
  BEGIN
2  For j=1 to $|\boldsymbol{P}|$
    BEGIN
3  Calculate the similarity $s(\boldsymbol{x_i},\boldsymbol{p_j})$
    END
4  Find the largest value $v$ of $s(\boldsymbol{x_i},\boldsymbol{p_j})$, for which the class of $\boldsymbol{p_j}$ is different from that of $\boldsymbol{x_i}$.
5  For each $s(\boldsymbol{x_i},\boldsymbol{p_j})$, add index j to $Q(\boldsymbol{x_i})$ if $s(\boldsymbol{x_i},\boldsymbol{p_j})>v$
  END
6 Let $\boldsymbol{P_{sel}} = \varnothing$
7 Find all $\boldsymbol{Q}(i)$ that have a single element. Add that element to $\boldsymbol{P_{sel}}$, and remove it from $\boldsymbol{P}$.
8 For all components of $\boldsymbol{Q}$, remove them if they have any element that is also in $\boldsymbol{P_{sel}}$
9 While there are any components remaining in Q do
  BEGIN
10  For all elements of $\boldsymbol{P}$, calculate how many times they appear in Q
11  Find the most occurring element, add it to $\boldsymbol{P_{sel}}$ and remove it from $\boldsymbol{P}$.
12  For all components of $\boldsymbol{Q}$, remove them if they have any element that is also in $\boldsymbol{P_{sel}}$
  END

Despite it's apparent complexity, this procedure is only $O(|\boldsymbol{P}| \times |\boldsymbol{X}|^2)$, and relies on two phases: calculating the Q-set (steps 1 to 5), and selecting the prototypes (steps 6-12). The Q-sets are simply a list of all prototypes in the same class as the given data pattern, and are closer than any prototype within a different class. The process of selecting the prototypes is simply a heuristic to solve a set covering problem.

## V. CLASSIFYING THE DATA

While Q-set minimization is more useful for high dimension problems, using a two dimensional one has the advantage of allowing us to visualize what is happening. In our case, we have a two dimensional problem. For each recording, we obtained a detection statistic by choosing the maximum of the set of quadratic detectors $\rho_S(t,f)$ on the time-frequency plane, for both the metallic and the rubber hammer cases. This provided two values, for each recording. We can thus graphically represent our data as seen in Fig.2.

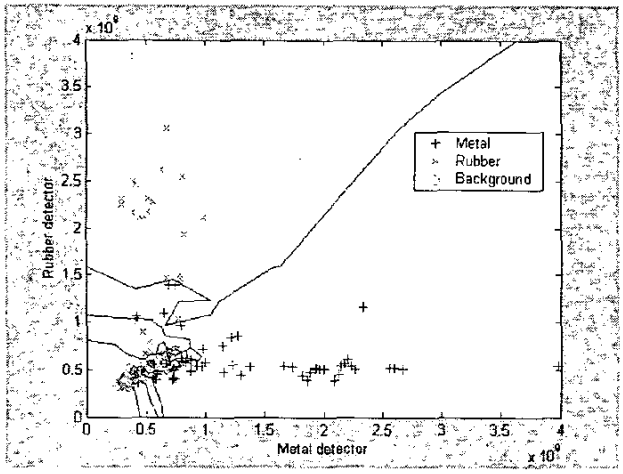


**Fig. 2 - 2D signature of collected data**

A strong overlap between the three classes clearly exists when the detectors yield low values, so a data editing technique is advisable. A commonly used data editing technique, originally proposed in [16], consists of removing from the data set any pattern for which the majority of it's $n$ nearest neighbors belong to a different class. This will sharply reduce the number of outlier prototypes, and remove data patterns from areas where another class has greater probability density.

The error rate on the training set using nearest neighbors is always zero, but we can have an estimate of the true error by using a leave-one-out cross validation [17]. For cross validation, one must divide the available data into a training set, that is used to design the classifier, and a separate test set, that will be used to estimate the error. When using the leave-one-out technique, only one pattern is removed to serve as test set in each experiment. We thus will have as many experiments as data patterns available.

When using leave one out cross validation, an error rate of 23.6.% (35 errors) was obtained, using 147 prototypes. Since, for each run, this technique will yield only a 0% or 100% error (only one test pattern is used each time), it does not make sense to assign a variance to this estimated error rate. The results can be seen in the following table

| Correct class / Given class | Metal | Rubber | Backgnd. |
|---|---|---|---|
| Metal | 45 | 3 | 10 |
| Rubber | 3 | 25 | 4 |
| Backgound | 10 | 5 | 43 |

If we apply the Q-set minimization, we will be able to use only an average of 44.4 ± 3.7 prototypes, and still obtain an error rate of 22.9%. This error rate is slightly better than the full nearest neighbor classifiers because, as discussed in [15], the simpler model will more likely have better generalization capabilities.

By applying the data editing technique described originally in [16] with a neighborhood of 3, we will remove 39 data patterns from the original set. If we now use the same leave-one-out technique to design the classifiers but use all the available data (including that removed by the editing process) for calculating the error rate, we will obtain an error rate of 17.4% ± 1.7 with the nearest neighbor classifier (using 108 prototypes). Using the Q-set minimization, we will obtain an error rate of 15.5% ± 1.5, using only 10.8 ± 1.1 prototypes. These values are summarized in the following table:

| Technique | Error rate | N. of prototypes |
|---|---|---|
| Nearest Neighbors | 23.6 | 147 |
| Q-set minimization | 22.9 | 44.4 ± 3.7 |
| Edited Nearest Neighbors | 17.4 ± 1.7 | 108 |
| Edited Q-set minimization | 10.8 ± 1.1 | 10.8 ± 1.1 |

As an example, we present in Fig. 3 the decision boundaries obtained with one of the Edited Q-set minimizations.

## VI. CONCLUSIONS

Model free, data driven methods are a workable approach to detection of underwater transients. The use of a time-frequency setup provides a useful covariance to doppler and delay shifts. However, the computational effort of the method is still considerably high. Kernel decomposition techniques do not seem to alleviate the problem, due to the fact that the kernel is data dependent. The linear separability assumption between transients does not hold even approximately, for the transients and background considered in this article. This forces the use of an additional classifier stage, where some care must again be taken, namely to alleviate the computational burden of working with too many prototypes. A technique to reduce the number of prototypes has been presented, and shown to be effective. By using this technique

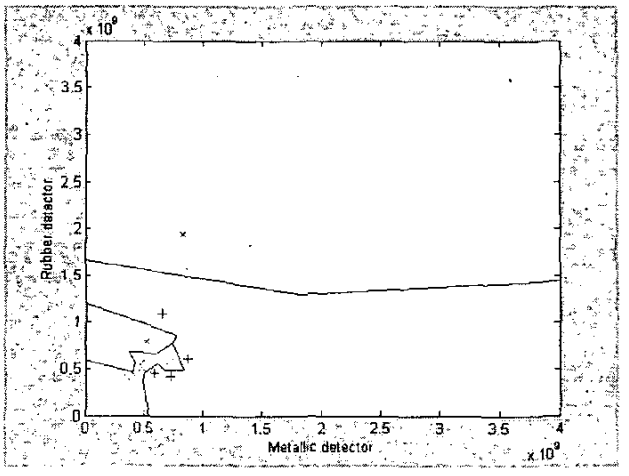


**Fig. 3 - Edited Q-set minimization**

(Q-set minimization), we are able to design a simple nearest neighbor based classifier that uses only 11 prototypes to classify the recordings into one of 3 classes, instead of the 147 original prototypes.

## Acknowledgments

The authors wish to thank Dr. Sebastien Bausson for many interesting discussions concerning some of the presented topics.

## References

[1] P. M. Oliveira and V. Barroso "Data Driven Underwater Transient Detection Based on Time-Frequency Distributions", Proc. of MTS/IEEE Oceans 2000, August 2000.

[2] D. L. Jones and A. M. Sayeed, "Blind Quadratic and Time-Frequency Based Detectors from Training Data, "*Proceedings of the 1995 IEEE International Conference on Acoustics, Speech and Signal Processing* – ICASSP 95, Detroit, MI, pp. 1033-1036.

[3] L. Cohen, *Time Frequency Analysis*, Englewood-Cliffs, NJ: Prentice-Hall, 1995.

[4] P. Flandrin , "A time-frequency formulation of optimum detection," *IEEE Trans. on Sig. Proc.*, vol. 36, pp. 1377-1384, Sept. 1988.

[5] C. S. Cunningham and W. J. Williams, "Fast implementations of generalized discrete time-frequency distributions", *IEEE Trans. on Sig. Proc.*, vol. 42, pp. 1496-1508, June 1994.

[6] C. M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press - Oxford, 1995.

[7] Duda, R. O., P. E. Hart, et al.. *Pattern Classification*, Wiley-Interscience, 2001.

[8] Fix, E. and J. L. Hodges. Discriminatory Analysis: Nonparametric Discimination: Consistency Properties. Randolph Field, USAF School of Aviation Medicine - Project 21-49-004: 261-272, 1951.

[9] Cover, T. M. and P. E. Hart. "Nearest Neighbor Pattern Classification." *IEEE Transactions on Information Theory* 13(1): pp. 21-27, 1967.

[10] Dasarathy, B. V. *Nearest Neighbor Pattern Classification techniques*, IEEE Computer Society Press, 1991.

[11] Kuncheva, L. I. and J. C. Bezdek. "Nearest Prototype Classification: Clustering, Genetic Algorithms, or Random Search ?" *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 28(1): pp. 160 –164, 1998.

[12] Aha, D.. "Lazy Learning - Editorial Review." *Artificial Intelligence Review* 11, 1997.

[13] Bax, E. "Validation of Nearest Neighbor Classifiers."*IEEE Transactions on Information Theory* 46(7): pp. 2746-2752, 2000.

[14] Nock, R. and M. Sebban. "An improved bound on the finite-sample risk of the nearest neighbor rule." *Pattern Recognition Letters* 22(3-4): pp. 407-412, 2001.

[15] Lobo, V., R. Swiniarski, et al. *Pruning a classifier based on a Self-Organizing Map using Boolean function formalization*. WCCI - World Conference on Computational Intelligence, Anchorage, Alaska, USA, IEEE Press, 1998.

[16] Wilson, D. L.. "Asymptotic Properties of Nearest Neighbor Rules Using Edited Data." *IEEE Transactions on Systems, Man, and Cybernetics* 2(3): pp. 408-421, 1972.

[17] Breiman, L., J. H. Friedman, et al., *Classification and Regression Trees*, Chapman & Hall, 1984.