NONSTATIONARY SIGNAL CLASSIFICATION USING SUPPORT VECTOR MACHINES

Arthur Gretton[1]*, Manuel Davy*[1]*, Arnaud Doucet*[2]*, Peter J. W. Rayner*[1]

[1] Signal Processing Group, University of Cambridge
Department of Engineering, Trumpington Street
CB2 1PZ, Cambridge, UK
{alg30,md283,pjwr}@eng.cam.ac.uk
[2] Department of Electrical and Electronic Engineering
The University of Melbourne
Victoria 3010, Australia
a.doucet@ee.mu.oz.au

ABSTRACT

In this paper, we demonstrate the use of support vector (SV) techniques, for the binary classification of nonstationary sinusoidal signals with quadratic phase. We briefly describe the theory underpinning SV classification, and introduce the Cohen's group time-frequency representation, which is used to process the non-stationary signals so as to define the classifier input space. We show that the SV classifier outperforms alternative classification methods on this processed data.

1. INTRODUCTION

The classification of nonstationary signals is a difficult and much studied problem. On one hand, the nonstationarity precludes classification in the time or frequency domain; on the other hand, nonparametric representations such as time-frequency or time-scale representations, while suited to nonstationary signals, have high dimension. Time-Frequency Representations (TFRs) and distance measures adapted to their comparison have previously been used to classify nonstationary signals [1, 2, 6], however the decision rules chosen in these studies limit the performance of these classification algorithms.

Support vector machines (SVMs) [10] provide efficient and powerful classification algorithms, which are capable of dealing with high dimensional input features, and with theoretical bounds on the generalisation error and sparseness of the solution provided by statistical learning theory [12, 10]. Classifiers based on SVMs have few free parameters requiring tuning, are simple to implement, and are trained through optimisation of a convex, quadratic cost function, which ensures the uniqueness of the SVM solution. Furthermore, SVM based solutions are sparse in the training data, and are defined only by the most "informative" training points.

In this paper, we propose to use a support vector machine for binary classification of the TFRs of nonstationary signals. In section 2, we review support vector classifiers. In section 3, we propose a classifier implementation based on Cohen's group TFRs, and in section 4 we compare the classification results obtained with the SVM-TFR approach to those found using other classification methods.

2. SUPPORT VECTOR CLASSIFICATION

We first describe how support vector machines may be used in binary classification, using the ν-SVR procedure. The results in this section are derived in Schölkopf *et al.* [9], and are also described in detail in Schölkopf and Smola [10].

Assume a sample of N labelled training points,

$$\boldsymbol{z} \triangleq ((\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)) \in (\mathcal{X} \times \mathcal{Y})^N,$$

in which $\mathbf{x}_i \in \mathcal{X}$, where $\mathcal{X}$ is the input space, and $y_i \in \mathcal{Y}$, where $\mathcal{Y}$ is the label space. For our purposes, we define $\mathcal{Y} \triangleq \{-1, 1\}$, which corresponds to a two class classification problem. We seek to determine a function

$$\begin{aligned} \psi : \mathcal{X} &\to \mathcal{Y} \\ \mathbf{x} &\mapsto \psi(\mathbf{x}), \end{aligned}$$

that best predicts the label y for a point $\mathbf{x}$. Assuming that random variable pairs $(\mathbf{x}, \mathrm{y})$ are generated i.i.d according to a distribution $\mathbf{P}_{\mathbf{x},\mathrm{y}}$, the optimal predicted class label for an input $\mathbf{x}$ is

$$\psi(\mathbf{x}) = \arg\max_{y} \mathbf{P}_{\mathrm{y}}(\mathrm{y}|\mathbf{x} = \mathbf{x}).$$

Since we do not know the mapping $\psi(\cdot)$, we define a learning algorithm $\mathcal{A}$,

$$\begin{aligned} \mathcal{A} : \bigcup_{N=1}^{\infty} (\mathcal{X}, \mathcal{Y})^N &\to \mathcal{H} \\ \boldsymbol{z} &\mapsto \psi_{\boldsymbol{z}}(\cdot), \end{aligned}$$

within a class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, which we call the *hypothesis space*, that is flexible enough to model a wide range of decision boundaries. We next define a *feature space* $\mathcal{F}$, endowed with an inner product[1] $\langle \cdot, \cdot \rangle_{\mathcal{F}}$, and a mapping from $\mathcal{X}$ to $\mathcal{F}$,

$$\begin{aligned} \mathbf{\Phi} : \mathcal{X} &\to \mathcal{F} \\ \mathbf{x} &\mapsto \mathbf{\Phi}(\mathbf{x}). \end{aligned}$$

Let us restrict $\mathcal{H}$ to functions of the form

$$\mathcal{H} := \{x \mapsto \text{sign}\left(\langle \mathbf{\Phi}(\mathbf{x}), \mathbf{w} \rangle + b\right) | \mathbf{w} \in \mathcal{F}, b \in \mathbb{R}\}.$$

We can then define a function $f_{\boldsymbol{z}}(x)$ in $\mathbb{R}^{\mathcal{X}}$, such that $\psi_{\boldsymbol{z}}(\cdot) = \mathcal{A}(\boldsymbol{z}) = \text{sign}(f_{\boldsymbol{z}}(\cdot))$; thus

$$f_{\boldsymbol{z}}(x) = \langle \mathbf{\Phi}(\mathbf{x}), \mathbf{w} \rangle + b, \tag{1}$$

and the problem of finding a *nonlinear* decision boundary in $\mathcal{X}$ has been transformed into a problem of finding the optimal *hyperplane* in $\mathcal{F}$ separating the two classes, where this hyperplane is parametrised by $(\mathbf{w}, b)$.

The mapping $\mathbf{\Phi}(\cdot)$ need never be computed explicitly; instead, we use the fact that if $\mathcal{F}$ is the reproducing kernel Hilbert space induced by $k(\cdot, \cdot)$, then

$$\langle \mathbf{\Phi}(\mathbf{x}_i), \mathbf{\Phi}(\mathbf{x}_j) \rangle_{\mathcal{F}} = k(\mathbf{x}_i, \mathbf{x}_j).$$

The latter requirement is met for kernels fulfilling the Mercer conditions [10]. These conditions are satisfied for a wide range of kernels, including Gaussian radial basis functions,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2_{\mathcal{X}}}{2\sigma^2}\right). \tag{2}$$

An estimate $f_{\boldsymbol{z}}(\cdot)$ associated with the *loss* $c\left(\mathbf{x}, y, f_{\boldsymbol{z}}(\cdot)\right)$ is attained by minimising the *risk* $R\left(g_{\boldsymbol{z}}(\cdot)\right)$, i.e.

$$f_{\boldsymbol{z}}(\cdot) = \underset{g_{\boldsymbol{z}}(\cdot) \in \mathcal{F}}{\text{argmin}} \left[R\left(g_{\boldsymbol{z}}(\cdot)\right) \triangleq \mathbf{E}_{\mathsf{x},\mathsf{y}}\left[c\left(\mathbf{x}, \mathsf{y}, g_{\boldsymbol{z}}(\mathbf{x})\right)\right]\right]. \tag{3}$$

Possible loss functions include the soft margin loss [3, 5],

$$c\left(\mathbf{x}, y, g_{\boldsymbol{z}}(\mathbf{x})\right) = \begin{cases} 0 & \text{if } y\, g_{\boldsymbol{z}}(\mathbf{x}) \geq \rho, \\ \rho - y\, g_{\boldsymbol{z}}(\mathbf{x}) & \text{otherwise,} \end{cases} \tag{4}$$

and the logistic regression loss [8],

$$c\left(\mathbf{x}, y, g_{\boldsymbol{z}}(\mathbf{x})\right) = \log\left(1 + \exp(-y\, g_{\boldsymbol{z}}(\mathbf{x}))\right), \tag{5}$$

among others. The present study is confined to the case of soft margin loss, which has been used successfully with support vector methods in a wide variety of classification problems [10].

In practice, equation (3) cannot readily be solved, as we do not usually know the distribution $\mathbf{P}_{\mathsf{x},\mathsf{y}}$. Minimising the empirical risk alone does not take into account other factors, such as the complexity of the classifying function, and can therefore result in overfitting [10, 12].

[1] We omit the inner product subscript in the subsequent discussion, unless the inner product is taken in a space other than $\mathcal{F}$.

We now describe the optimisation problem to be undertaken in finding $f_{\boldsymbol{z}}(\mathbf{x})$. All support vector classification methods involve the minimisation of a regularised risk functional, which represents a tradeoff between classifier complexity and training error (the latter is determined by the cost functional). In the case of the ν-SVR method, the regularised risk $R_{\text{reg}}(f_{\boldsymbol{z}}(\cdot), \boldsymbol{z})$ at the optimum is given by

$$\min_{f_{\boldsymbol{z}}(\cdot) \in \mathcal{F}} \left[R_{\text{reg}}(f_{\boldsymbol{z}}(\cdot), \boldsymbol{z})\right] = \min_{\mathbf{w}, b, \rho} \left[\frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + R^{\rho}_{\text{emp}}(f_{\boldsymbol{z}}(\cdot), \boldsymbol{z})\right], \tag{6}$$

where we use the soft margin loss from equation (4) in the empirical risk;

$$\begin{aligned} R^{\rho}_{\text{emp}}(f_{\boldsymbol{z}}(\cdot), \boldsymbol{z}) &= \frac{1}{N}\sum_{i=1}^{N} c(\mathbf{x}_i, y_i, f_{\boldsymbol{z}}(\cdot)) \\ &= \frac{1}{N}\sum_{i=1}^{N} \xi_i, \end{aligned}$$

in which

$$\xi_i = \max\{0,\ \rho - y\, f_{\boldsymbol{z}}(\mathbf{x})\}.$$

All training points $(\mathbf{x}_i, y_i)$ for which $y_i\, f_{\boldsymbol{z}}(\mathbf{x}_i) \leq \rho$ are known as *support vectors*; it is only these points that determine $f_{\boldsymbol{z}}(\cdot)$. The rôle of the term ν in equation (6) is described in the following theorem, from Schölkopf *et al.* [9].

Theorem 1 *The following results hold* only *for solutions to the optimisation problem in equation (6) for which* $\rho > 0$.

1. *ν is an upper bound on the fraction of training points for which $y_i\, f_{\boldsymbol{z}}(\mathbf{x}_i) < \rho$, which we call* margin errors.
2. *ν is a lower bound on the fraction of training points for which $y_i\, f_{\boldsymbol{z}}(\mathbf{x}_i) \leq \rho$ (the support vectors).*
3. *Assume a data set $\boldsymbol{z}$ generated iid according to $\mathbf{P}_{\mathsf{x},\mathsf{y}}$, and that neither $\mathbf{P}_{\mathsf{x}}(\mathbf{x}|\mathsf{y} = 1)$ nor $\mathbf{P}_{\mathsf{x}}(\mathbf{x}|\mathsf{y} = -1)$ contains any discrete component. Then, given a kernel $k(\cdot, \cdot)$ that is analytic and non-constant, with probability 1, asymptotically, ν is equal to the fraction of support vectors and the fraction of margin errors.*

It can be shown [9] that the component $\mathbf{w}$ in equation (1) is a linear combination of the mapped training points,

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{\Phi}(\mathbf{x}_i).$$

and that solving equation (6) is equivalent to finding

$$\max_{\boldsymbol{\alpha}} \left(-\frac{1}{2}\sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)\right)$$

subject to

$$0 \leq \alpha_i \leq \frac{1}{m},$$
$$\sum_{i=1}^{m} y_i \alpha_i = 0,$$
$$\sum_{i=1}^{m} \alpha_i \geq \nu.$$

There exist a number of methods that can be used to solve this quadratic programming problem. Our results were obtained using the Loqo algorithm in Vanderbei [11]. In the case of large training sets, data decomposition methods exist to speed convergence; see e.g. Chang *et al.* [4]. The offset b and soft margin loss parameter ρ are found using

$$y_j \left(\langle \mathbf{w}, \mathbf{\Phi}(\mathbf{x}_j) \rangle_{\mathcal{H}} + b \right) = \rho \quad \text{when } \alpha_j \in \left(0, \frac{1}{m}\right);$$

the set of equations thus obtained can be solved via linear least squares.

3. KERNEL DESIGN

The ν-SVM classification procedure relies on the choice of a kernel $k(\cdot,\cdot)$ suited to the problem at hand. In Davy *et al.* [6], a nonstationary signal classification algorithm was introduced, based on Cohen's group time-frequency representations. In this paper, we choose a ν-SVM kernel $k(\cdot,\cdot)$ based on a similar approach.

We write the Cohen's group time-frequency representation of $s(t)$ as $\mathcal{C}_s^{\phi}(t,f)$ (parametrised by its TFR kernel[2] ϕ). Given two signals $s(t)$ and $s'(t)$, the Gaussian radial basis function kernel of equation (2) then becomes

$$k(\mathbf{x},\mathbf{x}') = \exp -\frac{1}{2\sigma^2} \left[\int\int \left| \mathrm{N}\mathcal{C}_s^{\phi}(t,f) - \mathrm{N}\mathcal{C}_{s'}^{\phi}(t,f) \right|^2 dtdf \right], \tag{7}$$

where the notation $\mathrm{N}\mathcal{C}_s^{\phi}(t,f)$ is used to show that the TFR is normalised;

$$\mathrm{N}\mathcal{C}_s^{\phi}(t,f) = \frac{\left|\mathcal{C}_s^{\phi}(t,f)\right|}{\int\int \left|\mathcal{C}_s^{\phi}(t,f)\right| dtdf}. \tag{8}$$

In this formulation, the input space $\mathcal{X}$ defined in previous section is the space of normalised TFRs (i.e., $\mathbf{x} = \mathrm{N}\mathcal{C}_s^{\phi}(t,f)$), which depends on the choice of the TFR kernel ϕ.

4. RESULTS

We now apply the ν-SVR algorithm to the binary classification of chirp signals, and compare our results to those obtained previously by Davy *et al.* [6, 7]. The test signals are defined as the sum of two linear chirps:

$$\begin{aligned} x(k) &= A \sin\left[2\pi(a_0 + a_1 k)\right] \\ &+ B \sin\left[2\pi(b_0 + b_1 k + b_2 k^2)\right] \\ &+ \epsilon(k)\ ,\ k = 0, \ldots, K-1, \end{aligned} \tag{9}$$

where the $\epsilon(k)$ are iid, and are generated by a zero mean Gaussian process with variance σ_ϵ^2. Each test signal $x(k)$ is parametrized by $\theta = \left(A, B, a, b, \sigma_\epsilon^2\right)$, with $a = (a_0, a_1)$ and $b = (b_0, b_1, b_2)$. The problem consists of classifying a given signal $x(k)$ into one of the two following classes:

- Class ω_1 : $p(b_2) \sim \mathcal{U}(\frac{-0.30}{2(K-1)}, \frac{-0.20}{2(K-1)})$, where $\mathcal{U}(a,b)$ is the uniform distribution on (a,b),
- Class ω_2 : $p(b_2) \sim \mathcal{U}(\frac{-0.15}{2(K-1)}, \frac{-0.05}{2(K-1)})$.

The remaining signal parameters are identical in both classes, i.e. $A = B = 1$, $a_0, b_0 \sim \mathcal{U}(0,1)$, $a_1 = 0.25$ and $b_1 = 0.40$. The support of the noise-free time-frequency representation for signals in each class is plotted in figure 1.

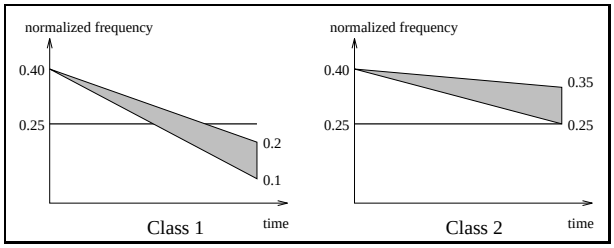

Fig. 1. Support of the noise-free TFRs (the gray areas represent the possible instantaneous frequencies for each class)

The ν-SVM algorithm was trained using 100 points, with an equal number of examples in each class. We specified a kernel width of $\sigma^2 = 0.1$ (see equation (7)), and set $\nu = 0.8$. A radially symmetric Gaussian TFR kernel ϕ was selected, with parameters optimised to minimise the error rate observed on the test data.

To measure the performance of the algorithm, a total of 20000 randomly generated test signals were used, again divided equally between the two classes (note that the training signals did not form part of the test set). Table 1 shows the average error over these test signals, compared with the average obtained over the same number of test signals for alternative classification methods. We see that for this problem, the ν-SVM algorithm achieves the lowest error rate.

5. CONCLUSION

In this study, we show that the good performance of SV classifiers in high dimensions allows us to effectively classify chirp signals, when these are transformed

[2] In order to avoid confusion between the ν-SVM kernel and the TFR kernel, the latter will be referred to as the *TFR kernel* at all times.

Classification method	Error rate
Wigner distribution [1]	22.30 %
Ambiguity plane [2]	4.56 %
Time-Frequency [6]	2.25 %
MCMC classification [7]	9.96 %
SVM classification [this paper]	1.51 %

Table 1. Error rates for the classification of chirps using the proposed SVM implementation and other classifiers.

using Cohen's group time-frequency kernels. Additional advantages of the SV classification method include simplicity of implementation, relatively low computational cost, and uniqueness of the SVM solution.

6. REFERENCES

[1] S. Abeysekera and B. Boashash. Methods of signal classification using the images produced by the wigner distribution. *Pattern Recognition Letters*, 12:717 – 729, November 1991.

[2] L. Atlas, J. Droppo, and J. McLaughlin. Optimizing time-frequency distributions for automatic classification. In *SPIE - The International Society for Optical Engineering*, 1997.

[3] K Bennett and O. Mangasarian. Robust linear programming discrimination of two linearly inseperable sets. *Optimisation methods and software*, 1:23–34, 1993.

[4] C.-C. Chang, C.-W. Hsu, and C.-J. Lin. The analysis of decomposition methods for support vector machines. *IEEE Transactions on Neural Networks*, 11(4):1003–1008, 2000.

[5] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, pages 273–297, 1995.

[6] M. Davy, C. Doncarli, and G. Faye Boudreaux-Bartels. Improved optimization of time-frequency based signal classifiers. *IEEE Signal processing letters*, 8(2):52–57, February 2001.

[7] M. Davy, C. Doncarli, and J.Y. Tourneret. Supervised bayesian learning using mcmc methods. application to the classification of chirps. Technical Report CUED/F-INFENG/TR.401, Engineering Department, University of Cambridge, UK, 2001.

[8] P. Huber. *Robust statistics*. John Wiley and Sons, New York, 1981.

[9] B. Schölkopf, A. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12:1207–1245, 2000.

[10] A. Smola and B. Schölkopf. *Learning with Kernels*. MIT press, To appear.

[11] R. J. Vanderbei. Loqo: An interior point code for quadratic programming. Technical Report TR SOR-94-15, Department of Civil Engineering and Operations Research, Princeton University, 1995.

[12] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, N.Y., 1995.