Proc. SPIE 1997, Volume 3162

Optimizing Time-Frequency Distributions for Automatic Classification

L. Atlas*, J. Droppo*, and J. McLaughlin**
*Department of Electrical Engineering
University of Washington
Box 352500
Seattle, WA 98195-2500

**S4-115
MIT Lincoln Laboratory
244 Wood St.
Lexington, MA 02173-9185

Abstract

An entirely new set of criteria for the design of kernels (generating functions) for time-frequency representations (TFRs) is presented. These criteria aim only to produce kernels (and thus, TFRs) which will enable more accurate classification. We refer to these kernels, which are optimized to discriminate among several classes of signals, as *signal class dependent kernels*, or simply *class dependent kernels*.

The genesis of the class dependent kernel is to be found in the area of operator theory, which we use to establish a direct link between a discrete-time, discrete-frequency TFR and its corresponding discrete signal. We see that many similarities, but also some important differences, exist between the results of the continuous-time operator approach and our discrete one. The differences between the continuous representations and discrete ones may not be the simple sampling relationship which has often been assumed.

From this work, we obtain a very concise, matrix-based expression for a discrete-time/discrete-frequency TFR which is simply the product of the kernel with another matrix. This simple expression opens up the possibility to optimize the kernel in a number of ways. We focus, of course, on optimizations most suitable for classification, and ultimately wind up with the class dependent kernel. When applied to simulated sonar transient signals, we find that our approach does a good job of discriminating within very similar classes of transients and is especially sensitive to differences in time variation across classes.

Introduction

Traditionally, kernel design and selection has been guided by the desire to have the resulting time-frequency representation (TFR) satisfy one or more established properties each of which is motivated by certain physical or mathematical considerations [1]. For example, we may wish for our TFR to be consistent, in some sense, with the time series and/or the Fourier transform of the time series. We may want our TFR to have the characteristics of an energy distribution. Or we may want our TFR to be shift-invariant, *e.g.* if we shift the time series, we observe a similar shift in the TFR. Not all TFR kernels will preserve all of these properties, but what all previously-proposed kernels lack are properties which relate directly to an end objective of signal classification.

In this paper we will be concerned with tasks where classification *is* the end objective. If TFRs such as those alluded to above are used, we have to hope that if our signal is well represented, then it will be well classified. Though this seems like a reasonable proposition, a representation may well bear a great deal of information which *all* signals under study share as well as information unique to each individual example signal. Such information is irrelevant to the classification, but most past approaches leave it up to a subsequent classifier to “screen out” these details.

Calculation of Discrete Time-Frequency Representations

Operators for time and frequency can be used together to produce time-frequency distributions. Unlike quantum physics however, construction of joint time-frequency distributions is hampered by a lack of one, unique Ψ function (which is dependent on both time *and* frequency) to describe a time-varying signal. This is due to an ambiguity in the ordering of exponentials in the characteristic function [3], as is reviewed below.

For continuous variables, we can calculate the characteristic function using the operator A via

$$M(\alpha) = \langle e^{j\alpha A} \rangle = \int_{-\infty}^{\infty} s^*(t)\, e^{j\alpha A} s(t)\, dt \tag{1}$$

from which the distribution in the a domain is calculated using

$$P(a) = \int M(\alpha)\, e^{-j\alpha a} d\alpha \tag{2}$$

(The $1/2\pi$ scale factor in front of the integral has been ignored here and in subsequent math.)

For a continuous joint time-frequency representation $P(t, \omega)$, the characteristic function is defined as

$$M(\theta, \tau) = \langle e^{j\theta T + jmW} \rangle \tag{3}$$

and the TFR can be calculated by performing a two-dimensional Fourier transform on the characteristic function:

$$P(t, \omega) = \iint M(\theta, \tau)\, e^{-j\theta t} e^{-j\tau\omega} d\theta d\tau \tag{4}$$

Since T and W are operators, the exponent in the characteristic function cannot be manipulated as if it contained ordinary variables. In fact, each different ordering of the operators in the exponential leads to a distinct characteristic function. For instance, $\langle e^{j\theta T + jmW} \rangle$, $\langle e^{j\theta T} e^{jmW} \rangle$, and $\langle e^{\frac{j}{2}mW} e^{j\theta T} e^{\frac{j}{2}mW} \rangle$ are three different characteristic functions yielding three different TFRs.

For a discrete joint time-frequency representation $P[n, k]$, we have a very similar situation. We can define the characteristic function as

$$M[\theta, m] = \langle e^{\frac{j2\pi\theta L}{N} + \frac{j2\pi mK}{N}} \rangle \tag{5}$$

and again we have different possible orderings for the characteristic function, for example $\langle e^{\frac{j2\pi\theta L}{N} + \frac{j2\pi mK}{N}} \rangle$, $\langle e^{\frac{j2\pi\theta L}{N}} e^{\frac{j2\pi mK}{N}} \rangle$, and $\langle e^{\frac{j\pi\theta L}{N}} e^{\frac{j2\pi mK}{N}} e^{\frac{j\pi\theta L}{N}} \rangle$.

We shall now use one of these characteristic functions and attempt to derive the discrete and continuous versions of a well-known TFR.

Rihaczek Distribution

Using the characteristic function $\langle e^{\frac{j2\pi\theta L}{N}} e^{\frac{j2\pi mK}{N}} \rangle$ (and omitting the $2\pi/N$ in the exponential in each of the equations

below), we have

$$M(\theta, m) = \langle e^{j\theta L} e^{jmK} \rangle = \sum_n x^*[n]\, e^{j\theta L} e^{jmK} x[n] \tag{6}$$

$$M(\theta, m) = \sum_n x^*[n]\, e^{j\theta L} x[n+m] \tag{7}$$

using the shift property. Analogous to equation 4 is:

$$P[n, k] = \sum_m \sum_\theta M(\theta, m)\, e^{-jmk} e^{-j\theta n} \tag{8}$$

Substituting equation 7 for the characteristic function,

$$P[n, k] = \sum_m \sum_\theta \left(\sum_u x^*[u]\, e^{j\theta L} x[u+m] \right) e^{-jmk} e^{-j\theta n} \tag{9}$$

Since $e^{j\theta L} = e^{j\theta n}$ in the time domain, we get

$$P[n, k] = \sum_m \sum_\theta \left(\sum_u x^*[u]\, e^{j\theta u} x[u+m] \right) e^{-jmk} e^{-j\theta n} \tag{10}$$

$$= \sum_m \sum_u x^*[n]\, x[u+m]\, \delta[n-u]\, e^{-jmk} \tag{11}$$

$$\sum_m x^*[n]\, x[n+m]\, e^{-jmk} = x^*[n]\, e^{jnk} X[k] \tag{12}$$

which is the discrete version of the well-known Rihaczek TFR [5]. This result is analogous to the continuous-time result which can be derived in a very similar fashion.

A New Formulation for Discrete TFRs

Using what we've learned from the operator theory approach, we can now write a new expression for discrete-time/discrete-frequency TFRs. This expression will serve as a jumping off point for exploring class-dependent representations in the next section on class-dependent kernels.

Rewriting equation 5 as a summation, we have

$$M[\theta, m] = \sum_n x^*[n]\, e^{\frac{j2\pi\theta L}{N} + \frac{j2\pi mK}{N}} x[n] \tag{13}$$

We know from our previous work that this expression is difficult if not impossible to simplify. However, we can make use of

the correspondence rule [2], which maps permutations of the terms in the exponent to a kernel function $\phi[\theta, m]$, which is a scalar function of θ and *m*. In other words, we can choose an ordering of the exponent which we know we can simplify, and then use different kernels to obtain all other distributions. We choose for our exponent the ordering (equation 6) which gave us the discrete Rihaczek distribution (equation 12).

$$M[\theta, m] = \sum_n x^*[n]\,\phi[\theta, m]\, e^{\frac{j2\pi\theta \boldsymbol{L}}{N}} e^{\frac{j2\pi m \boldsymbol{K}}{N}} x[n] \tag{14}$$

We wish to express equation 14 using only matrix notation. We must make sure to handle the exponential terms correctly in light of the fact that $\boldsymbol{L}$ and $\boldsymbol{K}$ are matrices. A correct way to exponentiate matrices is to first perform an eigenvalue decomposition on the matrix, and then exponentiate the elements of the diagonal matrix. $\boldsymbol{L}$ is already diagonal, so we simply exponentiate. $\boldsymbol{K}$, or more specifically, $\boldsymbol{K}_t$ (since it is the time domain operator we are dealing with here) is diagonalizable by $\boldsymbol{K}_f$. (See [9] for futher details.) Using the superscript *H* to indicate the conjugate transpose, we thus obtain

$$M[\theta, m] = x^H \phi \Lambda_1 \boldsymbol{F} \Lambda_2 \boldsymbol{F}^H x \tag{15}$$

where Λ_1 is the diagonal matrix $\exp\left(\frac{j2\pi\theta \boldsymbol{L}}{N}\right)$, $\boldsymbol{F}$ is the DFT matrix, and Λ_2 is the diagonal matrix $\exp\left(\frac{j2\pi m \boldsymbol{K}_f}{N}\right)$.

Taking the two-dimensional DFT of equation 15, we obtain the representation

$$[n, k] = x^H (\Phi \,.*\, \boldsymbol{F})\, \boldsymbol{F}^H \tag{16}$$

where the operator (.*) stands for an element-by-element product, and Φ is the two dimensional DFT of ϕ whose rows and columns have been have been cyclically shifted by *n* and *k*, respectively.

Advantages of the Discrete Approach

We have presented a framework within which both continuous-time/continuous-frequency TFRs *and* discrete-time/discrete-frequency TFRs may be generated without requiring any special sampling or treatment of the input signal to construct the discrete representation. This framework, an extension of existing work using continuous operators, produces discrete operators and discrete TFRs with many of the same properties and characteristics as in the continuous case. However, we have also seen that there are some clear differences between the continuous and discrete treatments. This should not be viewed as a shortcoming of our approach as we've dealt with both the continuous and discrete cases even-handedly. Rather, our approach reveals that the discrete case is not simply a sampled version of the continuous.

Most importantly, the discrete operator theory approach offers new analysis tools that enable us to not only analyze the properties of existing representations, but also to search for new representations. Our concise, matrix-based expression for TFRs (equation 16) will, as we shall see, enable the development of the class-dependent kernels as well as provide new insights into more familiar kernel properties.

Class-dependent Kernels

As we've said before, a single discrete signal is associated with an essentially infinite set of quadratic time-frequency representations (TFRs) through appropriate choices for the kernel function. Useful kernels can be selected by incorporating into the kernel design properties that are desired in the end representation.

Kernel design for quite a number of "desired properties" has been researched. Though some of these TFRs may offer advantages in classification of certain types of signals, they cannot hope to offer improved signal discrimination for all signals because discrimination is not one of the explicit goals of the kernel design procedure.

Here, we present a kernel design procedure in which signal discrimination (in the time-frequency plane) is the *only* goal.

Using our above operator theory formulation for TFRs, we are able to easily develop a closed-form solution for an optimally discriminating kernel which is not signal dependent, but *signal class* dependent. Use of TFRs for signal classification [6] and detection [7][8] (a similar problem) has been researched previously, but from the point of view of discovering which, if any, of the existing TFRs might succeed with certain signal types. The idea of custom designing kernels has been explored [4], but not with an eye to classification.

To develop our class dependent kernels, we begin with our expression for discrete TFRs which emerged from the operator theory work in the previous section. Unlike signal-dependent kernels, these kernels will depend not just on one signal, but on groups of signals with all members of each group or class being related in some fashion. Development of this kernel is not unlike the training of a classifier in that data, labeled by class, is set aside for an initial "learning" phase. In this case, the "learning" involves producing a kernel which results in TFRs which are maximally different from class to class. Once the kernel is developed for a particular set of training data, the kernel can be applied to "test" data. If the training data were good representatives of the classes, then TFRs developed with the test data should also be maximally different.

We will begin with the assumption that there are only two classes of data. This is a useful number of classes for several important problems, among them the problem of detection. We will discuss the issue of multiple classes, as well as demonstrating some results on actual sampled data, in subsequent sections.

Time-Frequency Plane

We wish to find a kernel such that the TFRs P_1 and P_2 for each of two signal classes are maximally separated:

$$\max_{\Phi} \left\{ \sum_{n,\,k} \left| P_1[n,k] - P_2[n,k] \right|^2 \right\} \tag{17}$$

(For the moment, we assume for simplicity that there is only one representative signal for each class.)

We begin our simplifications by noting that for a single ordered pair (n,k) and a given input signal x, the right-hand side of equation 16 is a linear combination of the elements of Φ. Let us write the coefficients of this linear combination as the column vector x_1. This vector contains the elements of the discrete Rihaczek distribution (equation 12). As n and k change, the rows and columns of Φ shift, but this shift can also be captured by rearranging the elements of x_1. So, by reshaping the Φ matrix into a vector $\overline{\Phi}$, the expression for a TFR can be rewritten as

$$P = \boldsymbol{X}^T \overline{\Phi} \quad \text{where} \quad \boldsymbol{X} = \left[x_1 \ \ldots \ x_N \right] \tag{18}$$

The $x_1, \ldots, x_N$ are column vectors containing the same elements, but in different orders. The elements of the representation $P[n,k]$ are now all contained in the vector P.

Having done this, we can rewrite equation 17 as

$$\max_{\overline{\Phi}} \left\{ \overline{\Phi}^H (\boldsymbol{X}_1 - \boldsymbol{X}_2)^H (\boldsymbol{X}_1 - \boldsymbol{X}_2) \overline{\Phi} \right\} \tag{19}$$

The expression in brackets is the quadratic form $y^H \boldsymbol{B} y$ which appears commonly in matrix algebra. It is maximized when y is the eigenvector corresponding to the largest eigenvalue of the matrix $\boldsymbol{B}$. Thus, we can calculate the kernel that produces maximal separation by performing an eigenvalue decomposition on $(\boldsymbol{X}_1\text{-}\boldsymbol{X}_2)^H(\boldsymbol{X}_1\text{-}\boldsymbol{X}_2)$.

Ambiguity Plane

For a signal of length N, the $\boldsymbol{X}$ matrix as given in equation 18 is of size $N^2 \text{x} N^2$. For a signal of any useful length, performing

an eigenvalue decomposition on a matrix of such size (even calculating the single eigenvector associated with the largest eigenvalue) is a lengthy process at best. Fortunately, we can circumvent this computation completely while obtaining the decomposition exactly by merely viewing the problem from the ambiguity plane, which is the two-dimensional Fourier transform of the time-frequency plane.

In the time-frequency plane, a TFR is computed from a kernel via a 2-D circular convolution, therefore the same TFR can be computed in the ambiguity plane via an element by element multiplication of the 2-D Fourier transforms of the matrices involved. Transforming equation 18 to the ambiguity plane then, we have:

$$A = \boldsymbol{Y}\Psi \tag{20}$$

A may be obtained from *P* by rewriting *P* as a matrix, taking the 2-D discrete Fourier transform and then revectorizing the result. The same method transforms Φ into Ψ, and the first column of $\boldsymbol{X}$ in equation 18 into the diagonal of $\boldsymbol{Y}$. $\boldsymbol{Y}$ is a strictly diagonal matrix, whose diagonal contains the elements of the 2-D discrete Fourier transform of the discrete Rihaczek distribution of the signal. Equation 19 can now be written in the ambiguity domain as:

$$\max_{\Psi} \{ \Psi^{H} (\boldsymbol{Y}_1 - \boldsymbol{Y}_2)^{H} (\boldsymbol{Y}_1 - \boldsymbol{Y}_2) \Psi \} \tag{21}$$

As $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$ are diagonal matrices, the eigenvalue decomposition is trivial in this ambiguity domain.

Looking at the kernel design in the ambiguity plane can also give us insight into what is actually being done. The kernel accentuates regions of maximum absolute difference (in the ambiguity function) of the Rihaczek distributions of the signals.

Multiple Classes and Multiple Class Representatives

Throughout this chapter, we have assumed a single representative signal for each class, but this constraint is not necessary. One way of incorporating multiple examples of each class in equation 17 is to average all the individual TFRs for class one and average all the TFRs for class two resulting in a representative P_1 and P_2 as is done in [6]. This is tantamount to averaging the $\boldsymbol{X}$ matrices in equation 18 or the $\boldsymbol{Y}$ matrices in equation 20. Equation 17 would then take the form

$$\max_{\Phi} \left\{ \sum_{n,k} \left| \frac{1}{N_1} \sum_{i=1}^{N_1} P_1^{(i)}[n,k] - \frac{1}{N_2} \sum_{j=1}^{N_2} P_2^{(j)}[n,k] \right|^2 \right\} \tag{22}$$

where N_1 and N_2 are the number of representatives for classes 1 and 2 respectively, and $P_x^{(i)}$ is the TFR for the *i*-th member of class *x*. This leads to the rewriting of equations 19 and 21 as

$$\max_{\Phi} \left\{ \Phi^{H} \left(\frac{1}{N_1} \sum_{i=1}^{N_1} \boldsymbol{X}_1^{(i)} - \frac{1}{N_2} \sum_{j=1}^{N_2} \boldsymbol{X}_2^{(j)} \right)^{H} \left(\frac{1}{N_1} \sum_{i=1}^{N_1} \boldsymbol{X}_1^{(i)} - \frac{1}{N_2} \sum_{j=1}^{N_2} \boldsymbol{X}_2^{(j)} \right) \Phi \right\} \tag{23}$$

$$\max_{\Psi} \left\{ \Psi^{H} \left(\frac{1}{N_1} \sum_{i=1}^{N_1} \boldsymbol{Y}_1^{(i)} - \frac{1}{N_2} \sum_{j=1}^{N_2} \boldsymbol{Y}_2^{(j)} \right)^{H} \left(\frac{1}{N_1} \sum_{i=1}^{N_1} \boldsymbol{Y}_1^{(i)} - \frac{1}{N_2} \sum_{j=1}^{N_2} \boldsymbol{Y}_2^{(j)} \right) \Psi \right\} \tag{24}$$

The problem solution then follows in the exact same way as was done before with only a small amount of additional computation needed to calculate the averages.

It will help to clarify the nature of the averaging being done here. Suppose we have three representatives of a class of upward-sloping chirps which are identical except that they differ slightly in their slope. The averaged TFR, as expressed by equation 22, is certainly *not* a TFR of a single chirp whose slope is, in some way, the average of the slopes of the three original chirps. Rather, the averaged TFR has amplitude everywhere that any of the original three chirps had amplitude. Moreover, the amplitude of the averaged TFR will be greatest at locations where the original three overlapped. This is a very desirable effect because we want our class-dependent kernel be computed using information from all of the representatives of all classes, and we also want the kernel computation to be weighted in favor of those time-frequency regions where many of our representatives have amplitude.

Though a number of important problems can be addressed using a two-class classifier, we would very much like to expand our approach to include the case of more than two classes. One way to do this is to simply extend the metric of equatin 17. Let us define the distance $D(P_i, P_j)$ as the very same distance between TFRs that we've defined previously in the two-class case

$$D(P_i, P_j) = \sum_{n,k} \left| P_i[n,k] - P_j[n,k] \right|^2 \tag{25}$$

Suppose that we have a three-class problem. Three different distances can be defined: $D(P_1, P_2)$, $D(P_2, P_3)$ and $D(P_1, P_3)$. A simple thing to do is to maximize the sum of those three distances, i.e.

$$\max_{\Phi} \{ D(P_1, P_2) + D(P_2, P_3) + D(P_1, P_3) \} \tag{26}$$

A big advantage to this strategy is that we end up with the very same sort of problem that we had before — a problem of the form $y^H \mathbf{B} y$ — which we know how to solve efficiently. This formulation also extends easily to the case of N classes by just summing up the distances for all possible pairs of TFRs.

The Data

Figure 1 shows a three typical time series from our data set. The data was intended to represent short duration, transient-like sonar acoustic events. Each signal was generated by tapping a 12-ounce glass bottle with a small hammer. Several classes of data were generated by filling the bottle with water in thirteen steps, adding about one ounce at each step. As the water level increased, it was hoped that the changing resonance of the bottle would make classification possible.

Many signals from each class were recorded on a digital audio tape. Later, the data was converted to analog and re-digitized at a sampling rate of 48 kHz. Thirty consecutive seconds of data from each class was digitized. All complete transients from each class were extracted with a 100ms window. This window was placed to contain 10ms of data followed by the entire transient.

When played at the original sampling rate of 48 kHz, it is possible for a human listener to make rough classification estimates. When played at a rate of 8 kHz, a human listener can discriminate between adjacent water levels (classes), but not with complete accuracy.

Three data sets were chosen for automatic classification, representing approximately eight, nine, and ten ounces of water in the bottle. Sixteen examples were available for each class. Half were reserved for testing data, and half were used as training data.

It was believed that these three classes would pose a significant problem for the classification algorithm.

Class Discrimination

Two classifiers were designed. The first classifier was based on our class-dependent kernel, described previously. The second classifier was based on the method of linear predictive coefficients (LPC) or, equivalently, an autoregressive model.

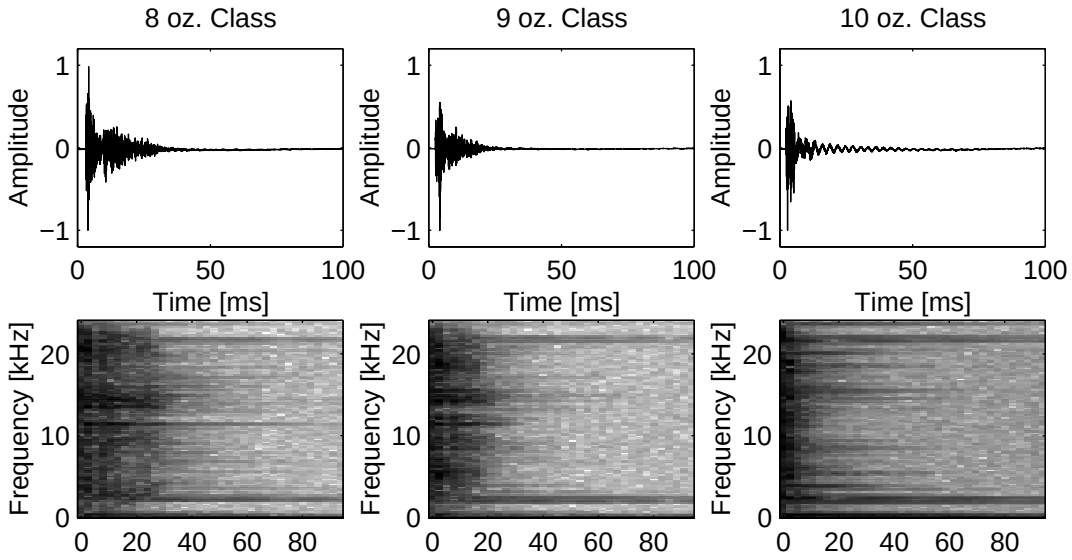

Figure 1: One example from each class, time series and spectrogram.

LPC Method

LPC are good at capturing the spectral content of a signal, but ignore any time-domain information that may be present. LPC assume that the observed signal was generated by passing either white noise or a periodic impulse train through a purely recusive discrete-time linear filter. The LPC are an approximation of the coefficients of this filter.

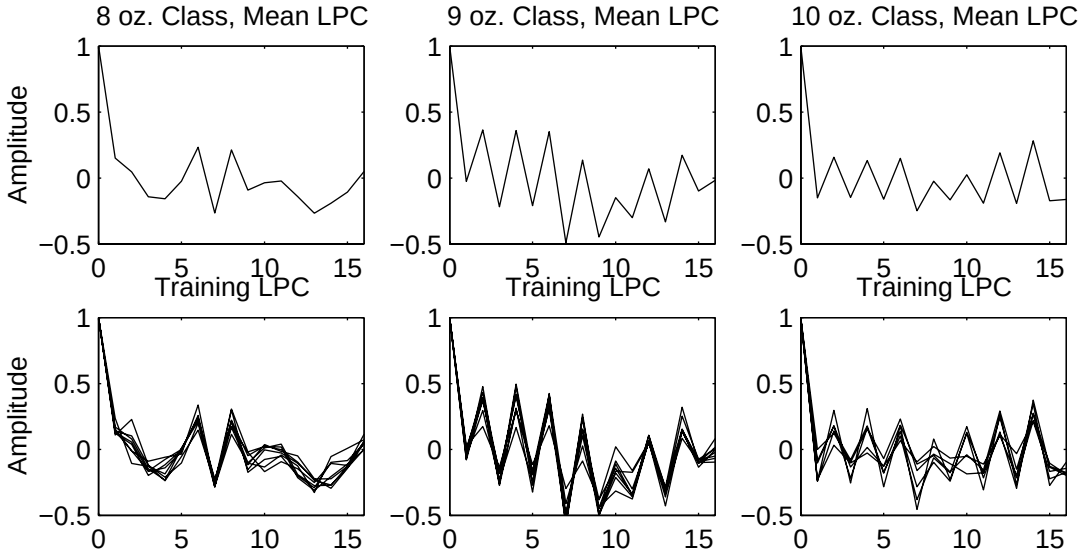

Figure 2: Linear predictive coefficients (AR(15))

For each training signal in each, the first sixteenf LPC were generated. Each class is then represented by its average LPC vector. Figure 2 shows the mean LPC vector for each of the three classes, along with plots of the vectors used for training.

Each signal in the test set was classified into one of the three classes by forming a new LPC feature, and determining which class mean was closest (in the Euclidean sense) to the test data.

The LPC classifier was able to perfectly classify all of the test data. We interpret this result as indicating that time-domain information was not relevant in the task of discriminating among our chosen classes. The spectral information is sufficient for classification in this way.

Class Dependent Kernel Method

Theoretically, our class dependent kernel should be able to find any time-variant behavior in the signals important for classification. The LPC contain only time-invariant spectral information, and do not have this luxury.

A kernel was generated to maximally separate the classes from one another. That is, we would expect our class-dependent kernel to generate distributions for each class which are far apart according to our Euclidean distance measure.

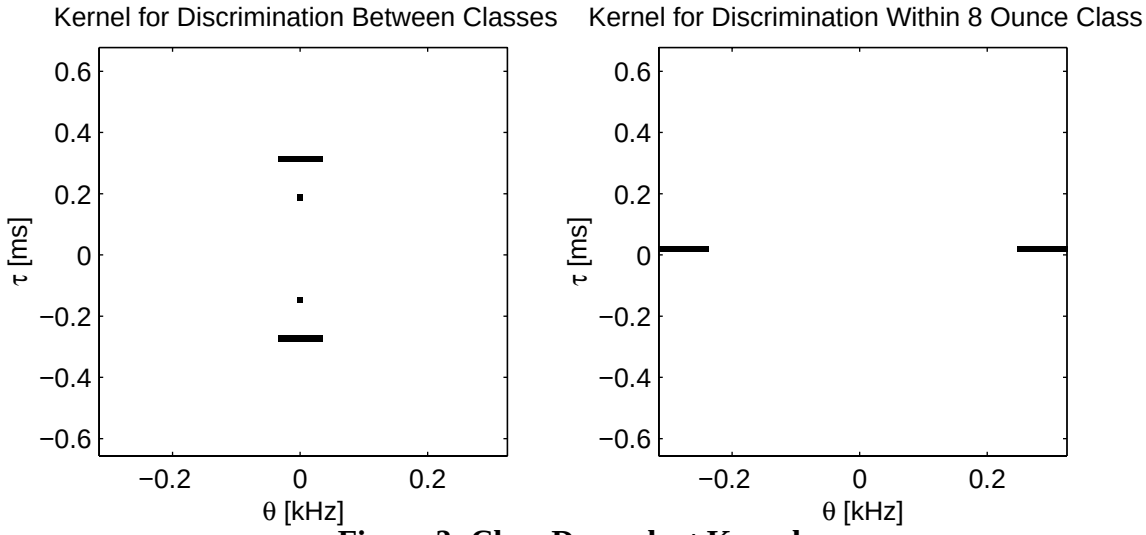

Figure 3: Class Dependent Kernels

The kernel found by our procedure is shown in the left half of Figure 3. By design, the kernel only takes on values of 1.0 and 0.0. The dark areas in the figure represent the only non-zero regions in the kernel.

This kernel focuses most of its energy along the $\theta = 0$ line in the ambiguity plane. Such a kernel tends to emphasize the spectral content of a signal. This indicates that, for this classification task, frequency information is more important. This agrees with our intuition that the difference between the classes is in the resonance of the bottle.

Figure 4 shows examples of Rihaczek distributions and class dependent distrubutions for one signal in each class. From these images, it is apparent that the kernel emphasizes the frequency differences while at the same time smooths the signals along the time axis.

Within Class Discrimination

Our class-dependent kernel method was used to generate a kernel that would optimally discriminate between signals within a single class. That is, the training data consisted of each signal from the training data for one class, separately.

The resulting kernel is reproduced in the right half of Figure 3. It has a concentration of energy along the $\tau = 0$ axis. In contrast to our previous example, this kernel tends to emphasize the time domain information in the signal. This indicates that within the class, the spectral information was uniform and not useful for classification. The relevant information exists mainly in the time domain. To summarize, the only within-class differences were the way the bottle was tapped from signal to signal, a difference which class-dependent kernels could, if desired, be designed to be sensitive to.

Conclusion

Using the concepts of operator theory, we've been able to forge a direct connection between a discrete, finite-length input

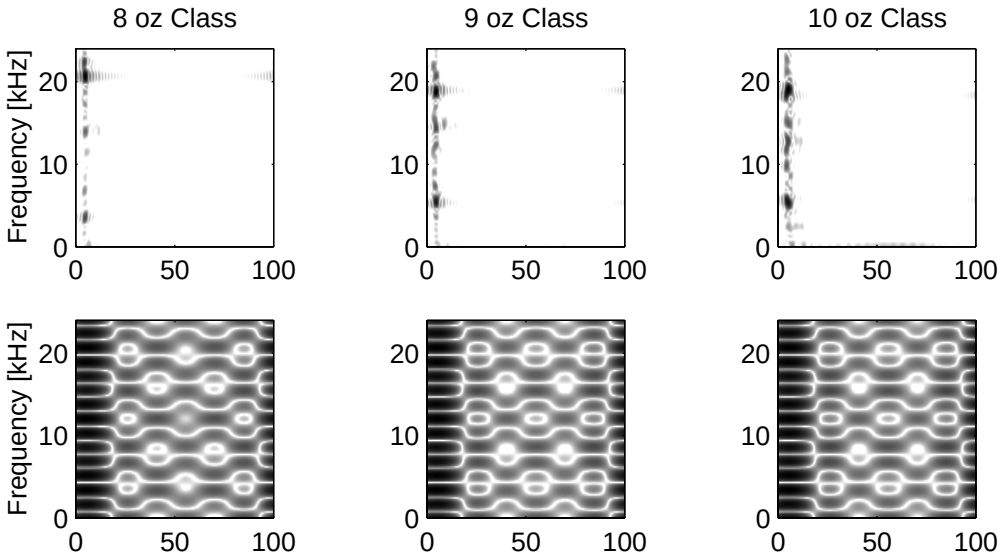

Figure 4: Example Rihaczek (top) and class dependent (bottom) distributions.

signal, and its discrete-time, discrete-frequency TFR. With our approach, we represent each TFR as the 2-D circular convolution of a kernel with the Rihaczek distribution of the signal.

It is important to note that the kernel we obtain for optimal separation maximizes the time-frequency difference *given* the original distribution (the Rihaczek). If the two signal classes have very dissimilar Rihaczek TFRs, then our method will find very little room for improvement.

Can a base representation other than the Rihaczek be used to form the $\boldsymbol{X}$ matrices in equation 19? Yes. Since one TFR can be derived from any other with application of the appropriate transforming kernel, any TFR may serve as an initial, base representation in our method. The optimal discriminating kernel will vary with the base TFR chosen, however, due to the varying amounts of time-frequency similarity between the signal classes.

In electing to use representative, "training" TFRs for each class in equation 17, we wind up with a kernel which, while maximizing the stated criteria for the training examples, is unlikely to maximize the distance between never-before-seen test examples. This is because different signals — even signals from the same class — will not have the exact same points of maximum difference (in the ambiguity plane) with signals from other classes owing to the stochastic variability of the process under study.

We have already mentioned the need to include multiple class representatives in the kernel design process if we are to be robust. But we will continue to be hamstrung in this area unless we include more points in our kernel than simply the points of maximum difference. In many applications involving eigenvector analysis, all eigenvectors associated with significant eigenvalues are utilized. We can do the same.

In choosing to include eigenvectors associated with eigenvalues other than just the largest ones, we no longer maximize equation 17. However, we may well optimize some other quantity when we consider the potential variability about the most different points in the ambiguity plane.

References

[1]T.A.C.M. Claasen and W.F.G. Mecklenbrauker, "The Wigner Distribution — A Tool for Time-Frequency Signal Analysis — Part I: Continuous-Time Signals," *Philips J. Res.*, vol. 35, pp. 217-250, 1980.

[2]L. Cohen, **Time-Frequency Analysis**, Prentice Hall Signal Processing Series, 1995.

[3]L. Cohen, “A General Approach for Obtaining Joint Representations in Signal Analysis. I. Characteristic Function Operator Method.” *IEEE Transactions on Signal Processing*, vol.44, no.5. pp. 1080-90, May 1996.

[4]M.G. Amin, G.T. Venkatesan and J.F. Carroll, “A Constrained Weighted Least Squares Approach for Time-Frequency Distribution Kernel Design,” *IEEE Transactions on Signal Processing*, vol. 44, no. 5, May, 1996.

[5]A.W. Rihaczek, “Signal Energy Distribution in Time and Frequency,” *IEEE Trans. Info. Theory*, vol. 14, pp. 369-374, 1968.

[6]I. Vincent, C. Doncarli and E. Le Carpentier, “Non Stationary Signals Classification Using Time-Frequency Distributions,” *Proc. of the Int. Sym. of Time-Frequency and Time-Scale Analysis*, p. 233-236, June, 1996.

[7]P. Flandrin, “A Time-Frequency Formulation of Optimum Detection,” *IEEE Transactions on Signal Processing*, vol. 36, no. 9, Sept., 1988.

[8]S. Kay and F. Boudreaux-Bartels, “On the Optimality of the Wigner Distribution for Detection,” *Proc. ICASSP 85*, pp. 1017-1020, 1985

[9]Jack McLaughlin and Les Atlas, “Applications of Operator Theory to Time-Frequency Analysis and Classification,” Submitted to *IEEE Trans. Signal Processing*, July, 1997