AUTOMATIC CLUSTERING OF VECTOR TIME-SERIES FOR MANUFACTURING MACHINE MONITORING

Lane M.D. Owsley[*], *Les E. Atlas*[*], *and Gary D. Bernard*[**]

[*]Interactive Systems Design Laboratory, Department of Electrical Engineering
University of Washington, Box 352500, Seattle, WA 98195-2500, U.S.A.
[**]Boeing Commercial Airplane Group, P.O. Box 3707, #M/S 5K-14,
Seattle, WA 98124-2207, U.S.A.

ABSTRACT

Our research in on-line monitoring of industrial milling tools has focused on the occurrence of certain wide-band transient events. Time-frequency representations of these events appear to reveal a variety of classes of transients, and a time-structure to these classes which would be well modeled using hidden Markov models. However, the identities of these classes are not known, and obtaining a labeled training set based on *a priori* information is not possible for reasons both theoretical and practical. Unsupervised clustering algorithms which exist are only appropriate for single vector patterns. We introduce an approach to unsupervised clustering of vector series based around the hidden Markov model. This system is justified as a generalization of a common single-vector approach, and applied to a set of vector patterns from a milling data set. Results presented illustrate the value of this approach in the milling application.

1. INTRODUCTION

In a traditional vector-sequence classification problem such as automatic speech recognition, it is possible to obtain training data which has been divided up into classes. The goal is to form models of these classes. As will be described in Section 2, however, we are working on an application in which class labels are not available. As a result, we need a method of determining automatically which natural groupings exist in the data set. A wide variety of research has been done in the area of unsupervised clustering[2], but the focus has been on clustering of patterns which consist of a single data vector. Our patterns are vector *time-series*, and forcing them into the single vector framework would be problematic and undesirable. We have developed a method of automatic clustering of vector sequences by generalizing a common vector-clustering method. Our new technique uses hidden Markov models (HMMs)[7] to define clusters, and we attempt to find the set of models which best describes the data distribution as a whole. We discuss the behavior of this algorithm and the influence of initial conditions, and illustrate the algorithm's successful use in our application.

2. CHALLENGES OF THE TOOL MONITORING APPLICATION

Our lab is conducting a major research effort into the evaluation of machining tool health based on the vibration patterns the cutters produce. Some of our recent work has been on milling tools. Evidence from a variety of sources [1],[3] has led us to focus on transient events which occur throughout the data set. We have observed that the frequency of these events changes throughout the life of the tool. It may be that transients occur *during* particular dulling events, or it may be that the transients occur *after* particular events (*e.g.* once a tool has chipped, it starts producing more transients). Some other transients may be totally unrelated to tool health. We have used time-frequency representations such as the spectrogram and other higher-resolution distributions[5] to observe that a wide variety of transients exist, but we have no *a priori* method of associating particular transients with particular event classes. We cannot ask a tool to "utter" a particular event class, even if we could determine exactly what all the event classes were. Furthermore, even on those relatively rare occasions in which we are able to establish that a particular dulling event occurred, we cannot conclude that a particular transient which occurred at that time is in fact related to the event. A transient may be of a type which occurs seemingly randomly throughout the data set, or a transient may be occurring as a result of a previous event. Our goal is to extract the transient types from the data, and then to relate the short and long-term trends in the frequency of particular types to known dulling events and to the overall health of the tool.

3.UNSUPERVISED CLUSTERING: VECTORS

Attempting to find classes which exist in unlabeled data is known as unsupervised clustering. This area has been intensively studied for situations in which each pattern consists of a single point [2]. Vector quantization (VQ)[4] is a form of unsupervised clustering; its objective is to represent a wide distribution of data vectors using a small

number of vectors (the codebook). If a VQ algorithm can find clusters which exist in the data and put its codevectors at the center of these clusters, then the codevectors will be better representations of the data than if the codevectors were sitting far from the bulk of the data. The Generalized Lloyd Algorithm (GLA)[4] is a VQ algorithm motivated by the recognition that an optimal quantization system for a particular data set meets two necessary (but not sufficient) conditions. One is that encoding be optimal given the codebook. The other condition is that the codebook be optimal given the partitioning of the training vectors. The codebook training procedure consists of an iterative application of these two constraints:

$$1: C_i[n] = \underset{j \in 1 \ldots J}{\operatorname{argmin}} \left\| \bar{b}_j[n] - \bar{t}_i \right\| \quad \forall (i \in 1 \ldots K) \quad (1)$$

$$2: \bar{b}_j[n+1] = \frac{\sum_{i=1}^{K} \bar{t}_i \delta_{i,j}[C_i[n] - j]}{\sum_{i=1}^{K} \delta_{i,j}[C_i[n] = j]} \quad (2)$$

where $\bar{t}_i$ is the *i*'th training vector, $\bar{b}_j[n]$ is the *j*'th codevector at iteration index *n*, and $C_i[n]$ is the classification result for the *i*'th training vector (i.e. the index of the codevector it is mapped to) at iteration *n*. Equation 1 says that, given the codebook $\bar{b}_j[n], j = 1 \ldots J$, optimally classify each of the *K* training vectors $\bar{t}_i, i = 1 \ldots K$ (by mapping them to the codevectors which are closest to them according to the L_2 norm). Equation 2 says that, given the classification results $C_i[n], i = 1 \ldots K$, optimally place the codevectors (at the centroid of the vectors which are mapped to them). These statements of optimality assume the L_2 norm classifier. The GLA has been shown to be useful in a variety of compression applications [4].

4. UNSUPERVISED CLUSTERING: VECTOR TIME-SERIES

VQ algorithms, though, are designed to work on *single*-vector data sets; our time-frequency representations are vector-*sequence* patterns. In theory, we could take the single-vector approach to our patterns by fixing the length of each transient and then forming one $K \times D$-dimensional "supervector" (*K*=number of time points; *D*=dimensionality of feature vectors). However, this destroys any time structure which may exist in a particular sequence and simply treats it as an unordered list of points. We instead use HMMs to represent clusters, because this allows us to express the time-structure information and because it is already the model we use in the classification stage [6].

5. OUR SEQUENCE CLUSTER REFINEMENT ALGORITHM

We have developed an algorithm, which we refer to as the sequence cluster refinement algorithm (SCRA), which is analogous to the GLA but which uses HMMs instead of template vectors to model clusters. The analogous algorithm for training HMMs based on data sets which were comprised of vector sequences is as follows: Given an initial set of *J* models $M_j[0]$ and *K* training sequences $t_i[k]$, iterate the following two conditions (with iteration index *n*:

$$1: C_i[n] = \underset{j \in 1 \ldots J}{\operatorname{argmax}} p\left(\bar{t}_i[k] \,|\, M_j[n] \right) \quad \forall (i \in 1 \ldots K) \quad (3)$$

$$2: M_j[n+1] = f\left(\bar{t}_i[k] : C_i[n] = j \right) \quad (4)$$

where $p(...)$ means probability and $f(\ldots)$ means "is a function of," where that function is in fact the HMM training algorithm. In step 1, we find the class labels $C_i[n]$ by determining which of our models $M_j[n]$ had the highest probability of producing the sequence $t_i[k]$. In step 2, we recalculate each model $M_j[n+1]$ by choosing it to maximize the probability of producing all the sequences which were most probably produced *by* it according to the first iteration. The result is that we are iteratively imposing two conditions which are necessary (but not sufficient) for the set of models which best describes the distribution of the overall data: the classification is optimal given the models, and the models are optimal given the classification.

6. PRACTICAL CONSIDERATIONS OF INITIAL CONDITIONS

The most important practical consideration of the algorithm which needs to be addressed is the choice of initial labels for the training sequences. The GLA is very dependent (in terms of final codevector locations, not compression performance) on initial conditions, and we have seen this to be true with the our clustering algorithm. However, we have also found that the importance of initial conditions is quite useful because it allows us to influence the selection of classes. We have approached this in two ways: by incorporating *a priori* information about what the classes *might* be or by incorporating information about what we would *like* the classes to be. The first approach was taken when we had knowledge of an existing event of interest (such as a chip being lost from the cutting edge of a tool) and we found a group transients in the region of that event. We would initially label those transients as members of the same class. The algorithm began by devoting one model to expressing the distribution of that group. As the iterations

progressed, the model refined its description of that class by passing off to other models patterns which didn't fit its distribution and attracting sequences which were good fits but which were previously members of another class (or unlabeled). The name SCRA acknowledges that our approach is in fact a process of refining the original class labels.

The second approach to initial class labeling is to choose class labels such that, if models could be found which produced similar labelings, the models would be useful in our application. For example, in our mill dullness monitoring application we began one training process by labeling a set of transients which occurred when the tool was sharp "Class 1," a group from the middle "Class 2," and a group from the dull region "Class 3." When our algorithm found models which produced labels which were as close to this starting point as possible, the models were useful to us as indicators of tool wear.

7. RESULTS

We have tested this algorithm extensively on real and artificial data, and in every test it has converged to a single state in eight or fewer iterations. Figure 1 illustrates the

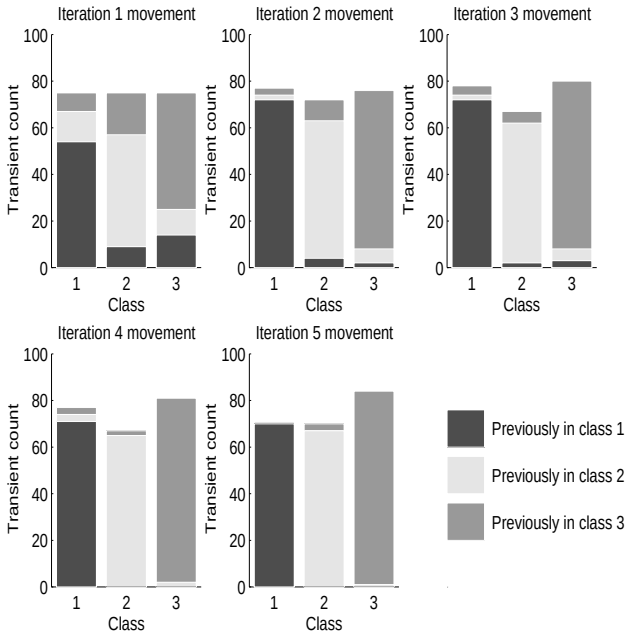

Figure 1: The evolution of transient class labels in a typical application of the sequence clustering algorithm. This is a three-model system; the bars indicate the classification results for the present iteration; the gray levels indicate which model the transients were used to train at the start of the iteration.

convergence of the algorithm for a typical data set. Each plot shows the change in labels for one iteration of the algorithm. It can be seen that in the first iteration approximately 1/3 of the transients are re-labeled. By iteration 5, only a few transients are relabeled. By iteration 6 (not shown), the class labels have solidified and all transients which were used to train a particular model are then best represented by that model; i.e. the algorithm has discovered distinct clusters.

We have looked at two methods of judging performance improvement due to the SCRA algorithm. The first looks at the value of classification results. In judging a traditionally trained classifier's performance, we would want to measure classification accuracy, but the motivation for the SCRA algorithm is that we don't know the correct classifications *a priori* and thus can't measure classification accuracy. As a substitute, we measure classification value with the assumption that if our classifier produces results which accomplish our goal (in this case labeling transients in a manner which is indicative of tool state), then the classifier must have identified "real" classes even if we don't immediately know what they are.

The method we have chosen to measure classification value is based around the relative entropy measure of the difference between two discrete distributions p and q:

$$H(p,q) = \sum_n p[n] \frac{\log p[n]}{\log q[n]} \tag{5}$$

We use this to compare the overall class distribution q to the local distributions p computed by dividing the overall transient sequence into smaller windows and finding the class distribution in each window. The result for a given experiment is a histogram of entropies, where higher entropy indicates that a given window is more different from the global distribution and thus the class labels provide us with more information useful in time-localization of the window in the data set as a whole, which is our goal. To calibrate these histograms, we create a second histogram by randomly creating a sequence of classes for the given global class distribution q. This histogram tells us what the probabilities of seeing windows of various entropies would be if there were no non-random structure to the class distributions as a function of time.

Table 1 summarizes some of the results according to this measure for 32 experiments conducted on milling data transients using a variety of signal representations and initial conditions. The table indicates that the SCRA algorithm was able to improve transient time-localization in virtually all of the cases. We have provided two measures: the percent of windows that exceeded the 5% threshold and the percent which exceeded the 1% threshold. The meaning of the 5% threshold is that, if there were no structure to the classification sequence other than that which could be expected to occur randomly, only 5% of the windows would exceed that threshold (the 1% threshold is defined similarly). According to both measures, there are a large number of

regions in which the class distribution is more greatly different from the global distribution than could be explained randomly, and the SCRA algorithm greatly increases that number. We are using these results in our research to focus in on those high-entropy regions of the tool's life and develop an understanding of what events occur there to make them different.

Table 1:

	5% threshold	1% threshold
% of experiments improved by SCRA	94%	100%
% of windows that exceeded threshold originally	16.68%	7.81%
% of windows that exceeded threshold after SCRA	22.94%	13.90%
% Improvement due to SCRA	37.53%	77.98%

Our second measure of performance was classification certainty. We have observed that, in every instance of our application of this algorithm, the final state was an improvement over the initial conditions in terms of fit of the models to the data. Figure 1 illustrates this for a set of 750

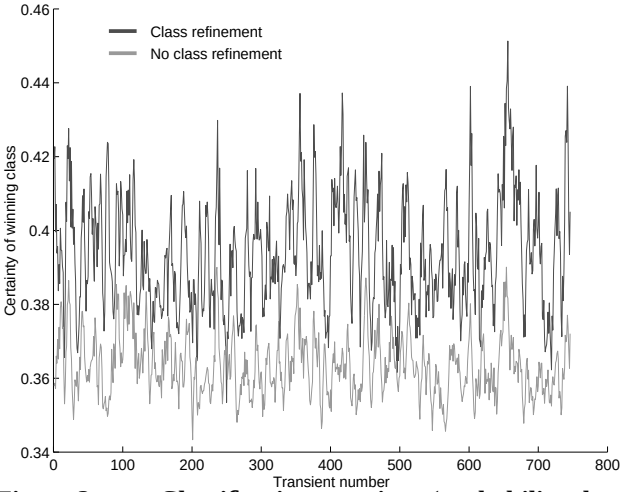

Figure 2: Classification certainty (probability that the chosen model was the correct one) before and after the use of our algorithm for a representative transient classification application.

transients taken from across the data set. For virtually all transients, the classification certainty has been improved.

8. CONTRIBUTIONS

- We have introduced a novel time-sequence unsupervised clustering algorithm which is based around the HMM, a cluster model specifically designed to express time-sequence information.
- We have justified this algorithm, with parallels to the successful GLA VQ algorithm, as an iterative application of two necessary conditions for optimal clusters.
- Our empirical studies have shown that the algorithm converges quickly and reliably, and produces valuable cluster estimates.
- We have shown how the use of initial conditions enables the incorporation of knowledge of known or desired class types.
- The strong applicability of this algorithm has been demonstrated in a key manufacturing application. This algorithm could also be used for applications in machine monitoring and speech recognition in which data is sparsely labeled or mislabeled.

9. REFERENCES

[1] Braun, S., Lenz, E., and Wu, C.L., "Signature Analysis Applied to Drilling," *ASME J. of Engr. for Industry*, vol. 104, pp. 268-276, 1982.

[2] Duda, R.O., and Hart, P.E., *Pattern Classification and Scene Analysis*. New York: John Wiley & Sons, 1973.

[3] Frarey, J. L., "Have We Finished?," *Sound and Vibration*, vol. 29, no 10, p. 5, 1995.

[4] Linde, Y., Buzo, A.,and Gray, R.M., "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84-95, Jan. 1980.

[5] Loughlin, P.J, Pitton, J.W., and Atlas, L.E. "An information-theoretic approach to positive time-frequency distributions," *Proc. ICASSP '92*, vol. 5, pp. 125-128, 1992.

[6] Owsley, L.M.D. and Atlas, L.E. "Machine-Tool Monitoring Using Time-Frequency Representations, Self-Organizing Feature Maps, and Hidden Markov Models," submitted to *IEEE Transactions on Signal Processing*, 1997.

[7] Rabiner, L.R., and Juang, B.H., "An Introduction to Hidden Markov Models,", *IEEE ASSP Magazine*, 4-16, Jan. 1986.