Trends & Perspectives In Signal Processing

Volume 2 Number 3 July 1982

KNOWLEDGE-BASED SIGNAL PROCESSING

Gary E. Kopec, Alan V. Oppenheim and Randall Davis

This work has been supported in part by the Advanced Research Projects Agency monitored by ONR under Contract N00014-81-K-0742 NR-049-506 at the Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, Mass. 02139. Reprinted with permission.

Commentary

Knowledge-Based Signal Processing

"Knowledge-based signal processing"—a term used to describe systems that tightly integrate artificial intelligence (AI) and signal processing—attempts to combine techniques from the two disciplines more imaginatively than in the past. Researchers in the signal processing community, in particular, are increasingly becoming aware that combining the architecture and methodology of AI with more traditional tools and techniques can lead to significant advances.

Previous Hybrid SP-AI Systems

The primary motivation for developing knowledge-based signal processing systems is the anticipated advantage from combining AI capabilities—symbol manipulation and knowledge representation—with the numerical and mathematical tools of signal processing. Of course, many systems already exist that incorporate and exploit principles from both fields, such as the Hearsay-II speech understanding system developed at Carnegie-Mellon University, the Surveillance Integration Automation Project (SIAP) system developed at Systems Control, Inc., of Palo Alto, Calif., and various image understanding systems. In general, any system that has a signal as its input and generates symbolic information as its output (or conversely, as in speech synthesis from text) must perform both signal processing and symbol manipulation.

Important limitations result from the way existing hybrid systems combine signal and symbol processing, however, and these limitations stem not from fundamental technical considerations but from historical, sociological, and management considerations. Previous efforts to merge AI and signal processing did not realize the full potential of the interaction because they tended to divide the signal processing and AI components of a problem along convenient but somewhat superficial lines. Past approaches to combining signal processing and AI have attempted to break down problems into separate signal processing and artificial intelligence components. In general, a large problem such as automatic surveillance or speech recognition must be partitioned into a number of smaller subproblems before it becomes tractable. Because separate communities have traditionally been responsible for advances in each field, the signal/symbol distinction is customarily selected as a convenient and well-defined dimension along which to factor large problems. This choice leads to an architecture in which the signal processing is localized in one set of subsystems, the symbol processing in another. The coupling between the two processors is typically weak, mirroring the coupling between the communities that developed them.

The specific manner in which the separate signal and symbol processing components interact is a related limitation. The most common paradigm is that of a signal processing front end that extracts features of an input signal and submits the results to a symbolic inference unit for further processing. In such a system, the information flow between the signal processing and symbol processing subsystems is typically unidirectional. The SIAP project, an attempt to use AI techniques to synthesize information from a variety of surveillance sources, illustrates the problems that can arise when a system is organized this way. Over the course of its development, SIAP went through three different incarnations, distinguishable by the three different signal processing front ends used. In the first version of the system, the signal processing took the form of an experienced human signal interpreter who manually fed to the symbol processing subsystem synthetic test data that had the same kinds of features the front end would ordinarily generate. In the next stage, a human interpreter scanned real acoustic data and provided the required signal characterizations. In the final version, the Signal Imagery and Management System (SIMS), independently developed at Bolt, Beranek, and Newman, Cambridge, Mass., replaced the human interpreter and provided a completely automated path from raw acoustic data to final interpretation.

The development of SIAP proceeded smoothly from the first stage, the synthetic scenarios, to the second stage, the use of manually transcribed data. When initially connected to SIMS, however, performance degraded considerably. The reasons were obvious. In manually transcribed data there were usually 40 to 50 signal events detected in each time interval. SIMS, on the other hand, typically provided about 300 signal events in the same time interval. Clearly, the human signal processors were doing considerably more than providing objective characterizations of the data. By relying on their experience, and by interpreting as they were processing, they were able to filter out 70% to 80% of what otherwise would have been called "events" in the data. Adjustments to the AI expertise in SIAP later enabled it to overcome some of the initial difficulties.

As the SIAP example illustrates, however, dividing a large problem into its "signal processing" and "artificial intelligence" components and addressing them separately may lead to unsatisfactory results. The prevalence of this approach stems, in part, from the adversarial relationship historically existing between the signal processing and AI communities. Selecting an approach to solving a signal interpretation problem has usually been phrased in either/or terms—rely principally either on sophisticated signal processing methods or on artificial intelligence techniques to provide most of the system capability. Indeed, the Defense Advanced Research Projects Agency (DARPA) speech understanding project was widely criticized for emphasizing the importance of syntactic and semantic (that is, symbol) processing at the expense of acoustic-phonetic (that is, signal) analysis. Provided that the techniques from the two disciplines can be effectively merged—and we think they can—there seems to be no fundamental reason to make such a drastic choice.

Comparisons Between Signal Processing and Artificial Intelligence

Although the need for more efficient interaction between signal processing and artificial intelligence is apparent, the exact form such interaction should take is not clear. Some basic difficulties are that signal processing systems traditionally work with numerical signals while AI systems are oriented more toward symbolic information and that the two disciplines exploit different kinds of knowledge about a task. Solving a signal processing problem usually involves formulating it in a way that makes the required solution mathematically apparent. A typical AI approach, on the other hand, identifies effective heuristics by focusing on how a human would solve the problem. The architectural styles of

VUS errors are more objectionable perceptually. Signal processing applications in which significant nonnumeric information is available about the signals being processed could also benefit from improved symbol manipulation capability. In some pitch detection and speech enhancement/restoration problems, for example, the text of the spoken utterance is available. This information can be very useful to a human who is performing manual pitch marking or listening to degraded speech, and it is likely that automatic systems could exploit it as well. In particular, an estimate of the number and order of voiced, unvoiced, and silent intervals might be derived from the speech text and used to guide VUS classification.

Knowledge Representation

Signal processing algorithms are typically based on mathematical models; the algorithms used in AI systems more often have a heuristic motivation. One manifestation of this is the difference in approaches to knowledge representation.

Signal processing system design is frequently dominated by a concern for run-time efficiency. As a result, a signal processing program is regarded primarily as a detailed specification of the sequence of operations involved in performing some computation. Program authors construct these specifications by applying their signal processing knowledge and insight to the task of accomplishing the desired goal, given some set of constraints and objectives. An important characteristic of the design process is that the relevant signal processing knowledge is identified and applied by the program author. The program itself contains only an indirect and implicit representation of the author's knowledge, in the form of the final algorithm.

In contrast, the problem of explicit knowledge representation is one of the most important and central areas of research within AI. In this field, a program is viewed not only as a specification of a computation, but also as a repository for the body of knowledge on which the computation is based. Indeed, one of the benefits that often accompanies the development of large knowledge-based systems such as PROSPECTOR, a mineral exploration system developed by D. Duda and P. Hart of SRI International, or MYCIN, a medical diagnosis system developed at Stanford University by E. Shortliffe, is the organization and formalization of the body of domain knowledge on which the program is based. Furthermore, AI programs are often intended to solve a variety of similar problems within some domain, rather than just one specific problem. The subset of knowledge appropriate in any particular situation is identified and applied not by the program author, but by the program itself.

Although there are many signal processing algorithms—the fast Fourier transform (FFT), for example—that can clearly be characterized as totally mathematical, it is far more usual to find algorithms motivated by both mathematical and heuristic considerations. This commonly occurs when a mathematically motivated algorithm is modified to correct deficiencies that first appear when it is tested on real data. The problem of pitch detection again provides a good example.

The development of a pitch detector typically begins with a mathematical model of periodic waveforms that suggests a way of measuring their period. Some of the measurements that have been used are the distances between the local extrema or zero-crossings of the time signal, the distances between autocorrelation or cepstral peaks, and the distances between peaks of the power spectrum. An algorithm based on such a model is tested on actual speech, noticed to work "most of the time," and perhaps modified to correct observed shortcomings. Unlike the initial formulation, these modifications are usually not motivated by a mathematical model of speech, but are introduced to handle real-world situations whose existence is suggested by the algorithm designer's experience. Typical of problems often handled in this way are pitch doubling and halving and hysteresis in detecting voiced-unvoiced transitions.

A second class of heuristically motivated signal processing algorithms are those in which the basic functionality to be implemented is defined in terms of the performance of a trained human expert, such as in geophysical signal interpretation, digital filter design (including choice of filter structure), target tracking, and satellite image interpretation. The tools that have been developed within AI for acquiring, representing, and applying expert knowledge appear to be particularly relevant in such applications.

Control Structures

The difference in knowledge representation between signal processing and AI programs corresponds to a difference in the kinds of program control structures that are used. While a typical signal processing program is a detailed prescription, explicitly specifying a sequence of operations, control flow in many AI programs is specified indirectly, in terms of a strategy rather than a detailed script. A control strategy specifies how to decide what to do at any given stage of processing, rather than spelling out what to do. Control strategy for a rule-based diagnosis system such as MYCIN might specify: "When the two disciplines also differ greatly. Signal processing programs tend to be relatively small, with algorithm design often dominated by considerations of run-time efficiency. Large AI systems, on the other hand, are among the most complicated programs written, and considerable effort is directed toward designing software that is easy to develop and maintain.

These differences suggest that close and effective interaction between AI and signal processing must begin with a clear understanding of the relationships between them and of the relative merits of each approach. Attempting to contrast the two along a number of basic dimensions—information representation, knowledge representation, control structures, data modeling, and the treatment of uncertainty in decision making—will help clarify the potential contributions each discipline can make to a "knowledge-based" signal processing system. It will also suggest specific ways advanced signal processing applications can benefit from increased collaboration with AI.

Information Representation

The ways in which discrete-time signals are represented in typical signal processing and AI programs provide a good illustration of the differences between the numerical and symbolic techniques.

Mathematically, a one-dimensional discrete-time signal might be defined as a real-valued function on some set of integers. As functions, two signals are equal if they take on the same value at each point of their domain. A signal that is the output of a sine-wave generator, for example, is equal to the signal that results from passing the output of a cosine-wave generator of the same frequency through a quarter-cycle delay. If only the numerical values of their samples are observed, there is no way to distinguish between the two signals.

In general, all signal processing algorithms are based on the concept of signal equality. This observation has been exploited in the design of SPL, a general-purpose, signal processing language developed by G. Kopec. In SPL, signals defined on finite intervals are characterized in terms of two basic operations, length and fetch. The length operation returns the length of the interval on which the signal is defined; the fetch returns the value of a specified sample. These operations are intended to provide a sufficient framework for building a powerful, general signal processing programming system.

A typical symbolic approach to signal representation, on the other hand, would augment these observable attributes to include explicit information about how the signal was created. Such information would allow a sine wave to be distinguished from a phase-shifted cosine wave, even when the two signals were mathematically equal as functions. These additional signal operations return information about the actual history of the signal; they are not estimation operators that decide whether the signal could have been created by a sine or cosine generator.

Thus the numerical approach to signal representation deals with the signals themselves, as abstract, mathematical functions; the symbolic approach deals with symbolic descriptions of signals. Because, in general, a given mathematical signal may be described in a variety of ways, a descriptive signal representation may result in an element of ambiguity. On the other hand, a description of how a signal was created represents a richer sort of information than does knowledge of its sample values. An "intelligent," interactive signal processing system, which avoids redundant computation by maintaining a data base of signals that it has computed, is a simple example of an application for descriptive signal representation. Whenever the user or a program requests a signal, the data base is examined to see whether it contains a signal description equivalent to that of the requested signal. If an equivalent description is found, the corresponding signal is returned. Otherwise, an appropriate procedure is invoked to create the desired signal and install it in the data base. In subsequent requests for the signal (or a signal that is recognizably equivalent to it) the stored value will simply be returned. Obviously, for such a signal cache to be useful, it must cost less, on the average, to locate an equivalent signal description than to numerically compute a signal value.

Although numerical and symbolic techniques are usually associated with different classes of problems, many signal processing applications require both kinds of capability, as in the problem of speech pitch detection. The voiced/unvoiced/silence (VUS) classification, which is normally considered part of the speech pitch detection task, is an instance of a computation that produces symbolic output; but the estimation of specific pitch values for voiced regions is a numerical process. (The fact that a zero pitch value is often used to represent unvoiced speech does not change the fact that VUS classification is fundamentally a symbolic reasoning process.)

The more difficult aspect of pitch detection appears to be the symbolic VUS classification task. VUS classification accuracy tends to degrade more rapidly than numerical pitch accuracy in the presence of background noise. Moreover, in many applications VUS classification accuracy is the stronger determinant of overall system acceptability. This is true in speech bandwidth compression, for example, where there is a choice for the next rule to try, use the one that is most likely to contradict the current hypothesis."

With programs that use a control strategy rather than a fixed control regimen, the specific sequence of operations that occurs during any execution of the program is not predetermined; rather it depends on the details of the problem being solved. Consequently, the program can adapt its processing to the characteristics of the input data. The arsenal of available operators might include some that are very effective for certain classes of input but totally inappropriate for others. By always choosing the best alternative operator, the system can perform well over a diverse input space. Furthermore, applying each operator only when there are specific reasons to believe it will be effective makes it feasible to accommodate a larger range of inputs than if each operator were applied in every situation. The availability of alternative operators may also contribute to increased robustness in the presence of noise or other anomalies in the data. The acoustic-phonetic analysis of speech is one particularly attractive application for such a control strategy. A basic problem in phonetic analysis is that no small set of acoustic measurements is appropriate for uniformly describing all possible speech sounds. Furthermore, a given phoneme may have a variety of different acoustic manifestations, depending on contextual factors such as phonetic environment, speaking rate, and stress. Nevertheless, it is likely that any single realization of a particular sound can be identified on the basis of a small subset of the whole collection of possible features. This is exactly the kind of situation in which context-dependent operator selection is likely to be advantageous.

Using a control strategy rather than a fixed script also helps maintain a separation between the static "competence" aspects of a system (what it knows, as reflected in the arsenal of available operators) and the dynamic "peformance" aspects (how it applies that knowledge, as reflected in the rules for operator selection). If this distinction is maintained, at least during system development, the functional adequacy of a system can be established before its run-time efficiency is optimized.

Data Modeling

Signal processing and AI use different kinds of models in data interpretation problems. Typically, raw data are obtained by "viewing" the "real world" through a transducer of some sort. The process of interpretation involves building a model of the data, of the transducer that generated the data, or of the underlying system. Signal processing models typically refer to the data, while AI models refer to the underlying system or transducer that generated the data.

The way each discipline approaches problems in image interpretation illustrates the difference. Digital image processing focuses on the image as a two-dimensional array of samples. This perspective is clearly apparent in traditional approaches to tasks such as image enhancement, restoration, and coding. Similarly, signal processing approaches to image analysis and scene classification are based on pattern recognition, using features (edge orientation, texture) measured directly from the image. Yet the description of a given object is strongly affected by a variety of nonintrinsic factors such as a viewpoint, occlusion by other objects, and orientation. This is a fundamental problem for such iconic approaches.

A significant portion of the AI work in image understanding, on the other hand, has focused not on the images themselves but on creating symbolic structural descriptions of the three-dimensional objects that generated the images. One motivation for this approach is that human visual reasoning appears to be done in terms of inferred three-dimensional object structure rather than immediate two-dimensional appearance. A second motivation is that the description of an object in terms of its three-dimensional structure is independent of contextual factors like those mentioned above. Thus, tasks such as object detection should be easier using symbolic scene descriptions rather than image features.

To assess the relative merits of these two approaches, we must balance the cost of obtaining a three-dimensional object description with the difficulty of obtaining an adequate characterization of all possible two-dimensional image manifestations of each object. In highly constrained contexts, image processing approaches appear to be significantly more cost-effective at present. This balance is likely to change, however, as the need for more general and robust image analysis systems grows and VLSI technology becomes more readily available.

Identifying the advantages and limitations of signal processing and AI approaches to image modeling is relatively easy because the problem has received serious attention from both disciplines. Other aspects of image processing, such as enhancement and restoration, have been exhaustively studied by those in the signal processing field but have received little attention from the AI community. Since there is a growing consensus that substantial additional progress in these areas is unlikely to come from signal processing alone, investigating the possible application of AI techniques to these tasks seems to be a direction with considerable potential.

Signal modeling problems involving sound could also benefit from the application of AI techniques developed in vision research. A computational model of hearing, analogous to current computational models of vision, might form the basis for new techniques of sound localization, sound classification, and speaker separation. As in vision research directed at problems such as stereopsis (for example, the work of D. Marr and T. Poggio at M.I.T.), an important aspect of the development of a computational hearing model would be identifying an appropriate class of primitive features for describing sound structures. Such primitives would be analogous to the "intrinsic images" (H. Barrow and J. Tenenbaum, SRI International) and "2.5-dimensional sketches" (D. Marr, M.I.T.) of early vision research.

Uncertainty in Decision Making

In signal processing, decision making in the face of uncertainty is usually formalized in probabilistic terms. Typical approaches to detection and estimation, for example, involve supplying a priori information in the form of multivariate conditional probabilities. If a large number of variables are involved, reliably estimating these probabilities can be difficult and may require analysis of a vast amount of data. The probability estimation problem is particularly acute in the case of infrequent events. It has been a source of difficulty, for example, with speech recognition systems like the one under development at IBM that are based on stochastic models.

Knowledge-based AI systems, on the other hand, typically embody a subjective concept of uncertainty rather than an objective, probabilistic one. Experts providing the knowledge base for a program such as PROSPECTOR or MYCIN, for example, are asked to supply the strengths of the inference rules that link the hypothesis with relevant evidence. Effectively, the knowledge and judgment of the expert, accumulated through experience, are substituted for unknown conditional probability functions. A major problem with this subjective approach is the possibility of inconsistency in the assignment of likelihoods.

When adequate probability estimates are available, a formal statistical technique is usually the method of choice. However, since there are many situations in which adequate data are not available, statistics alone is not a general practical methodology for dealing with uncertainty. When formal methods are impractical or only partially applicable, signal processing researchers typically resort to various *ad hoc* decision-making procedures. Compared with approaches based on objective probability alone, subjective techniques are practical over a much larger class of problems. Furthermore, compared with various combinations of probability and *ad hoc* decision-making methods, they provide a more uniform and formal methodology for dealing with uncertainty.

—Gary E. Kopec, Alan V. Oppenheim, Randall Davis

Gary E. Kopec and Alan V. Oppenheim are members of the Editorial Board. Randall Davis, assistant professor of computer science at the M.I.T. Artificial Intelligence Laboratory, is currently active in research on expert systems, knowledge acquisition, and applications of artificial intelligence to signal processing.