An Experimental ECG Scanning System

Kenneth P. Birman, Joseph Touch

Department of Computer Science
Cornell University, Ithaca, New York 14853

ABSTRACT

Our previous work on ECG “Holter” Scanners suggested that a *rule-based learning* approach would yield a more automatic and more accurate computerized holter scanning system. We have now built such a system, focusing on a number of pragmatic objectives: it must run on widely available hardware (we based our work on a SUN 3 workstation), provide a simple to understand and easily used operator interface that exploits window-oriented graphics capabilities of this machine, and run automatically except during the editing phase. An important subgoal was to implement an ST-segment measurement capability as part of the system. A non-goal was to “prove the superiority of the rule-based approach,” which is fortunate because the pragmatic requirements of the system have lead us to abandon some aspects of this approach, although retaining others. The software is not yet completed, hence it is not yet possible to evaluate its performance in detail. However, it is already clear that the system works extremely well. It is highly accurate, tolerant of noise and far faster and easier to use than an ECG analysis system we developed at Columbia University some years ago. In a collaboration with Columbia University and NASA’s Johnson Space Center, the system will be used experimentally at Columbia later in 1987.

I. AN OVERVIEW OF THE SYSTEM

Our system is structured as follows. Data is collected using a playback unite and copied to disk with a resolution of 12-bits/sample and a sampling rate of 250 data points/second. A preliminary data reduction phased reduces the signal to a “triangle” representation that closely mimics the original motion of the signal relative to baseline, while reducing data volume by a factor of 20-25. An adaptive-threshold technique preserves sensitivity to both high and low amplitude events without requiring backtracking, and because there is no eye-closing mechanism, associated anomalies do not occur. On the other hand, there may be many outputs per QRS complex.

The signal is scanned in a single automated pass. A QRS detection algorithm operates on the triangle encoded signal representation as follows. Maintaining the best candidate seen in a given interval of the signal, triangles are examined on both channels in temporal order. As each triangle is considered, either the current candidate is accepted as a “valid qrs”, or the candidate is discarded in favor of a “better” one. However, the QRS acceptance mechanism itself looks back at the previous output, and may under some conditions discard a detected QRS as apparent noise, for example if it finds a normal QRS with a normal RR interval and the event was of an unknown morphology or only appeared in a single channel. The decision as to how good a candidate any given complex might be is done in part by searching a table of known triangle patterns, thus biasing the program in favor of event morphologies that have been seen before. All such information is dynamically maintained and reflects both channels of the signal. We have observed that as the scan progresses, the likelihood of a QRS being classified correctly indeed rises.

Once QRS candidates have been identified, a morphological classification is performed, using a primitive but surprisingly robust method: the complex is ranked 0-1 in each of [a] set of attributes – big in the top channel, big in the bottom one, deflection is up, deflection is down, or both, event is “very” large, etc. the ranking is encoded into a 12-dimensional “binary” shape space, in which all coordinates are either 0 or 1. Within this large (internal) shape space, there are 4096 possible shapes, and a count is formed giving the number of events of each shape. Next, these initial categories are merged into at most 10 QRS morphology class[es] as follows. We add to the count for “shape” *i* the counts for all the adjacent shapes, found by changing the value of any single bit or any two bits in s. these merged counts are sorted, and up to 10 QRS morphologies are then assigned by first taking the most populated shape and its neighbors, then considering the next most populated one, etc. The 10^{th} morphology is a catch-all category covering everything left over after the basic assignment is done. We find that the method reliably discriminates morphologically different events while requiring minimal computer time and no operator time. A preliminary rhythm classification is performed on the classified waveforms. Noise is identified by applying a threshold to the frequency of triangle outputs from the encoder.

The operator becomes involved during the next phase of the analysis, during which the signal is edited. The program first computes a tabular summary of major morphological events and arrhythmias detected by the system. It sorts this table by event class and, within each class, by severity of the detected event. The operator is then presented with a menu describing the important “events” in this summary and the morphological classifications that were computed during the first phase (see Figure 1). The operator edits the output of the system by merging morphological classes, correcting QRS delineation for representatives of the normal classes, and examining successive arrhythmia events until the most extreme classifications have either been validated, corrected, or relabeled as noise. One minute of 2-channel data is displayed on the screen at a time.

The operator has complete freedom to review a different event category, skip an event category, and to redelineate or change the interpretation of the events shown on the screen. For example, to relabel a PAC complex as a PVC, one uses the mouse to point to it and depresses ‘V’ (or ‘N’ for noise, ‘delete’ to erase it, etc.). A magnifying glass mode can be used when fine-grained examination of the signal is needed, but most editing takes place at low resolution, and a typical editing operation requires only that the mouse be pointed at an event and a single mouse button or key depressed. The operator can also manually direct the program to display a certain section of the signal, for example in order to correlate electrocardiographic events with log entries or medication.

Rhythm categories include PVCs, interpolated PVC’s, PVC pairs, bigeminy, PAC’s, tachycardias of all types, bradycardia, and ST-segment depression or elevation. The operator can spend as much or as little time as desired in each category, focusing on accuracy in those aspects of a recording that are most important to the study in question, or just performing a cursory review if the recording is for clinical purposes and accuracy is less of an issue. When an editing change requires it, rhythm reclassification is automatically performed. In addition, the event table is recomputed as needed.

Figure 2 illustrates an interaction with the system in its magnifying glass mode.

During the initial phase, a printed report is generated. The report generator has been designed but not implemented. Like the scanner, it will be a graphical, menu-driven program, permitting the operator to edit textual parts of the report and to re-arrange ECG "strips" into any desired order, modifying the annotations if necessary and deleting strips if the report is longer than desired. The operator can edit *any part* of a report, although we certainly hope that numeric quantities will only be manipulated infrequently and for good reason (for example, to avoid the overhead of re-editing the entire signal just to correct some minor mistake). The final report is then spooled for output to a laser printer.

II. Learning about rule-based learning

The most difficult and critical components of any ECG analysis system are the preprocessing and QRS detection/classification stages. Errors made at these stages of the analysis must be corrected manually, if at all, and may recur again and again, perhaps even leading to a systematic to misclassify events. For example, many commercial systems are observed to miss all PVCs of certain shapes, or mislabel them repeatedly, or miss PVC pairs and tachycardias. Although arguably acceptable in the analysis of clinical holters, such behavior can clearly bias research results in ways that could subtly influence the outcome of a study or conclusions about the efficacy of a medication.

Our basic premise when we set out to build this new system was that adaptivity of the lowest levels of the analysis could be used to overcome any tendency to commit systematic errors. In particular, our goal was to design a preprocessing and analysis phase that could learn from its experience and thereby avoid making the sorts of errors that arise when a system applies a general rule naively in a situation to which it does not apply [Birman-a] [Birman-b].

Prior to arriving at the version of our analysis algorithm described above, we designed and implemented a preliminary system that actually had to be taught QRS shapes before it could recognize them. The idea was to detect only "known" complexes, and, after missing an event, to request that the operator explain to the system what had happened. Eventually, when the likelihood of encountering new QRS and PVC morphologies was felt to be low enough, the system would switch t a free running mode. The system learned complexes using the rule-based learning methodology described in [Birman-a]. We found that the resulting scanner was highly accurate even on signals with a wide variety of recurrent QRS and PVC morphologies and highly variable QRS amplitudes.

Rapidly, however, it became evident that the operator interaction required by this approach was excessive. Moreover, one stage of the rule-based methodology, "conflict resolution" (which is applied when two interpretations look equally plausible but are in conflict, say because they involve overlapping segments of the signal) required an inordinate amount of operator interaction. It became clear that to be useful the system would have to be much more automatic. We first tried to retrofit automatic conflict resolution and self-teaching modules into thee system, but eventually concluded that this strategy was backwards. The QRS analysis algorithm described above resulted from a redesign of the code as it existed at the end of this previous stage; it retains elements of the rule-based learning method (this is used to bias the detector in favor of events that have been seen previously), but uses a simple reliable algorithm, described above, to formulate large numbers of hypotheses about the signal, pick he best one, and then move on. The algorithm was easy to develop and debug, runs at very high speeds (nearly 240 times realtime on our SUN 3/160), and rarely commits systematic errors except in the delineation of QRS complexes. This last problem, which is common in commercial scanners, is overcome by an operator-assisted redelineation feature in the second stage analysis, prior to ST-segment measurement.

III. Conclusions

We have reported on the status of research "in progress" on a new electrocardiographic analysis system. Although we are not yet prepared to present detailed performance figures, nor have we gained experience with the system in actual use, preliminary indications are extremely encouraging. A year from now we hope to have both experience and data to present.

IV. Acknowledgements

Our previous work on ECG signal processing was done in collaboration with Dr. J.T. Bigger of Columbia University, to whom we are deeply indebted. The work is being done privately by the principle author and is not supported by any outside funding source.

V. References

[Birman-a] Birman, K.P., Rule-based learning for more accurate ECG analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (July 1982).

[Birman-b] Birman, K.P., Using SEEK for multichannel signal processing. *Computers and Biomedical Research* (August 1983).