

# ECG의 자동진단을 위한 컴퓨터 알고리즘

안재봉 · 윤형로\* · 이명호

= Abstract =

## Computer Algorithm for the Automatic Diagnosis ECG

Chae-Bong Ahn, Hyung-Ro Yoon\*, Myoung-Ho Lee

This paper presents a practical approach to develop computer algorithm for automatic diagnosis electrocardiogram (ECG) based on a decision table techniques.

### 1. 서 론

최근 컴퓨터를 이용한 의료진단은 광범위하게 적용되어 오고 있으며 이것은 의사와 환자 모두에게 많은 도움을 주고 있다. 특히 의사에게는 수동적인 해석 방법보다 시간을 크게 절약할 수 있으므로 많은 양의 진단과 치료를 신속하고 정확하게 행할 수 있다는 중요한 장점을 제공하게 된다. 더구나 장래에는 임상 자료의 급속한 증가가 예상되므로 이에 대한 효과적인 진단처리를 위하여 컴퓨터의 사용은 필연적이 될 것이다.

과거에 다변수 판단 분석법이나 또는 Bayes의 조건 확률 이론 또는 이 두가지 이론을 결합한 의료진단기법이 이미 발표된 바 있으나, 이러한 방법은 (1) 순수한 수학적 공식에 의해 표현되었거나 또는 (2) 의학자료로 적용되지 않는 독립적인 증상(symptoms)에 근거를 두었기 때문에 수학기식이나 확률통계에 사용되는 용어에 익숙치 못한 의사들에게는 실제 적용에 있어서 매우 제한되어 있는 실정이다.

따라서 본 연구에서는 자동적으로 의료진단을 행하는데 있어서 더욱 실제적이고 효과적인 접근을 하기 위해 판정테이블(decision table) 기법에 의한 컴퓨터 알고리즘을 제안해 보고자 한다. 원래 판정테이블은 상용 및 과학기술에 관련되는 자료를 처리하기 위해 오래전 부터 사용되어 왔다. 의료진단에 판정테이블의

적용은 진단 과정이 계산적이라기 보다는 주로 논리적이라는 사실로부터 비롯되었고 증상과 질병 사이의 많은 복잡하고 논리적인 관계를 표현하는데 있어서 이상적인 방법이 될 것이다.

따라서 본 논문에서는 ECG의 진단을 위한 판정테이블을 설계하고 이 판정테이블에 의해 자동 진단할수 있는 컴퓨터 프로그램을 개발하고자 한다.

### 2. 판정 테이블

#### 2-1. 판정 테이블의 구성

판정 테이블은 그림 2-1 과 같이 수평과 수직복합을

DECISION TABLE FORM		
	CONTENT	ENTRY
		RULE NUMBER
C O N D I T I O N		
	EACH ROW CONTAINS A CONDITION TO BE TESTED	EACH COLUMN IS A PARTICULAR COMBINATION OF TESTS TO BE SATISFIED
A C T I O N		
	EACH ROW LISTS AN ACTION TO BE TAKING	EACH COLUMN CONTAINS ACTIONS DICTATED BY TEST COMBINATION SATISFIED ABOVE

그림 2-1. 판정 테이블 구조

Fig. 2-1. The structure of a decision table.

<1982. 11. 29 접수>

연세대학교 공과대학 전기공학과

\* 연세대학교 원주대학 의공학과

Dept. of Electrical Engineering, Yonsei University

\* Dept. of Biomedical Engineering, Wonju College,

Yonsei University

이용하여 네개의 영역으로 나뉘어 진다. 즉 수직복선으로는 오른쪽의 “entry(항목)”와 왼쪽의 “content(내용)”를 구분하며 수평복선으로는 “조건(condition)”과 “실행(action)”을 구분한다. 즉 그림 2-1의 왼쪽상단은 시험되어질 일련의 조건이 기록되며 왼쪽하단은 이들 시험조사의 결과에 따라 나타나는 질병의 명칭이 기록된다. 한편 그림 2-1의 오른쪽 상단 엔트리의 각 열(column)은 한개의 판정규정(decision rule)을 표시하며 각 판정규정은 그림 2-2의 네 엔트리중의 어떤 하나로 구성된다.

Entry	Definition
“Y” means YES	this condition is to be tested to see if it is true.
“N” means NO	this condition is to be tested to see if it is true.
“ ” means BLANK	this condition does not apply, or this action is not to be taken when this rule is satisfied.
“X” means X	this action is to be taken when all conditions for this rule are satisfied.

그림 2-2. 판정규정의 정의  
Fig. 2-2. Definition of a decision rule.

판정 규정은 그림 2-1에서 볼 수 있는 것처럼 편의상 연속적인 숫자로 나타냈으나 이것은 판정 규정이 이처럼 연속적으로 수행되어야만 한다는 것을 의미하는 것은 아니다. 만일 각 판정 규정이 유일한 경우로 판정되어 다른 규정이나 혹은 연속적으로 나타나는 판정규정에 무관하다면 이것은 가장 이상적인 판정이 된다. 각 판정 규정에 의하여 규정된 일단의 조건들은 애매 모호한 판정을 피하기 위하여 유일한 조건이어야 한다. 동일한 조건으로 서로 다른 판정을 내리게 하는 것은 오진의 원인이 된다. 그러나 이와는 반대로 서로 다른 조건으로 동일한 판정을 내리게 하는 것은 가능하며 바람직한 경우다. 판정테이블에서 특별히 규정하지 않았거나 또는 의미가 없는 규정들은 ELSE 규정의 적용을 받으며 이것은 판정테이블의 조건이 기록되는 란에 아무런 기재사항이 없는 오른쪽 열을 말한다. 또한 조건 수, 질병의 수 그리고 규정의 수에는 제한이 없으나 될수록 작은 판정테이블을 여러개 적당히 연결하는 것이 편리하다.

판정테이블의 규정번호의 수는 제한되지 않으며 수

행될 조건과 나타나는 진단의 종류도 제한되지 않는다. 판정테이블에서 사용되는 명령문은 하나의 판정테이블에서 또다른 판정테이블로 옮겨가는 “GO TO” 명령문과 동일한 판정테이블 내에서 행과 열만을 바꾸어 가는 “DO” 명령문의 두가지가 있다. 즉 “GO TO” 명령문을 사용하면 판정테이블의 한 진단과정의 규정번호에 해당되는지의 여부가 확인된 후 다른 판정테이블로 이동될 수 있다. 판정테이블 사이의 연결명령문을 사용함으로써 하나의 업무를 여러 부분으로 나누어 각 부분에 대하여 판정테이블을 만들 수 있다. 또한 “DO” 명령문을 사용하면 한 판정테이블 안에서 다른 행과 열의 판정을 수행한 후 본래의 판정테이블로 되돌아가게 하면 남아있는 진단의 내용이 수행된다.

2-2. 판정테이블에 의한 진단

그림 2-3은 ECG로 부터 심장의 전도장애(conduction defect)를 진단하기 위하여 구성된 판정테이블이다.

그림 2-3의 판정테이블에서 조건란의 첫번째 열은 두개의 사지유도(limb leads) 전극으로 측정된 QRS 주기가 0.12sec와 크거나 같은지를 조사한다. 이 시험에서 “YES”라는 결과는 판정규정 NO.1, NO.3, NO.5, 및 NO.7에 나타나며, “NO”라는 결과는 판정규정 NO.2, NO.4, NO.6 및 NO.8에 나타난다. 이제 판정규정 NO.3에 대해 고찰해 보자. “YES”로 나타나는 요소는 판정규정 NO.3의 제1, 3 및 제6번째 조건을 만족시키며 전혀 응답이 나타나지 않는 조건은 제2번째와 제7번째인데 이 조건들은 판정규정 NO.3에는 적합치 않음을 뜻한다. 한편 제4번째와 제5번째 조건에 서는 “NO”라는 응답이 나타났는데 이것은 위의 조건

	DECISION TABLE FOR		TABLE NAME		RULE NUMBER							
	DIAGNOSING ECG CONDUCTION DEFECTS		CONDUCTION DEFECTS		1	2	3	4	5	6	7	8
C O N D I T I O N S	QRS DUR > 0.12 SEC IN ANY 2 LIMB LEADS				Y	N	N	N	N	N	N	N
	QRS DUR > 0.12 SEC IN ANY 2 LIMB LEADS				Y	N	N	N	N	N	N	N
	PR INTERVAL > 0.12 SEC IN ANY 2 LIMB LEADS				Y	Y	Y	Y	Y	Y	Y	N
	INTRINSIC QD DEFLECTION ONSET > 0.06 sec IN I, AVL, V5, V6				Y	N	N	N	N	N	N	N
	INTRINSIC QD DEFLECTION ONSET > 0.05 sec IN I, AVL, V5, V6				Y	N	N	N	N	N	N	N
A C T I O N S	COMPLETE LEFT BUNDLE BRANCH BLOCK		X									
	COMPLETE LEFT BUNDLE BRANCH BLOCK		X									
	COMPLETE RIGHT BUNDLE BRANCH BLOCK			X								
	INCOMPLETE RIGHT BUNDLE BRANCH BLOCK			X								
	COMPLETE ATYPICAL BUNDLE BRANCH BLOCK				X							
	INCOMPLETE ATYPICAL BUNDLE BRANCH BLOCK				X							
	WOLFF-PARKINSON-WHITE SYNDROME											X
	GO TO TABLE INSELECTION PART I		X	X	X	X	X	X	X	X	X	N
END OF CHECKING											X	

그림 2-3. 심장의 전도장애 판정테이블  
Fig. 2-3. Decision table for conduction defects of the heart.

과 다르다는 것을 의미한다. 따라서 판정규정 NO.3에 해당되는 응답은 왼쪽 응답란의 제3행에 나타난 것과 같이 “complete right bundle branch block”과 8행에 나타난 “GO TO table infarction part I”이다. 그러므로 판정규정 NO.3는 “만일 2개의 사지유도전극으로부터 측정된 QRS 주기가 0.12sec 와 크거나 같고, PR 간격이 0.12sec 보다 크거나 같으면, 그리고 I<sub>1</sub>, AVL, V<sub>5</sub> 및 V<sub>6</sub> 중 어떤 유도전극에서 측정된 intrinsicoid deflection onset이 0.045sec 보다 크지도, 같지도 않으면, 그리고 V<sub>1</sub>이나 V<sub>2</sub> 유도전극으로 측정된 intrinsicoid deflection onset이 0.06sec 보다 크거나 같으면 complete right bundle branch block이라는 진단이 내려지게 되며 아울러 infarction part I table의 검사가 명명된다.

진단도구로서 판정테이블의 이용은 다음과 같은 방법으로 수행된다. 즉 환자로부터 얻어지는 임상데이터를 판정테이블의 기록된 조건과 비교한다. 예를 들면 환자로부터 ECG를 측정된 결과 QRS 주기가 0.10sec, PR 간격이 0.14sec, intrinsicoid deflection onset V<sub>6</sub> 유도전극에서 0.03sec, V<sub>1</sub> 유도전극에서는 0.05sec로 되었다고 하자. 이때 전도장애 테이블에 있는 조건에 대하여 검사되어진 항목을 생각해 보면 QRS의 주기가 0.10sec 이기 때문에 첫번째 판정규정 NO.1의 조건은 만족되지 않으므로 두번째 판정규정 NO.2로 이동된다. 이 두번째 판정규정에서는 제1, 2, 3 및 제4번째 조건이 만족되며 제5번째의 조건은 맞지 않으므로 다시 제3번째 판정규정 NO.3로 이동된다. 그러나 환자의 ECG 데이터는 판정규정 NO.3에도 일치하지 않으나 마침내 판정규정 NO.4에 만족되므로 이 판정규정 NO.4가 가리키는 질병의 명칭은 제4행과 8행에 나타나 있는 것으로 판정된다. 따라서 환자의 임상데이터에 일치하는 첫번째 판정규정을 만나면 더 이상 오른쪽 판정규정으로 진행되지 않고 모든 판정규정을 무시하게 된다. 만일 판정테이블의 어떠한 판정규정도 만족되지 않는다면 ELSE 규정이 응답으로 나타나게 되며 table infarction part I으로 이동된다.

### 3. ECG 자동진단 컴퓨터 알고리즘

#### 3-1. 판정테이블의 컴퓨터 프로그램

판정테이블을 컴퓨터에 사용하려면 판정테이블을 컴퓨터가 읽을 수 있는 코오드로 변형시켜야 한다. 컴퓨터 프로그램으로 판정테이블을 변화시키는 첫번째 단계는 두개의 2진행렬식(binary matrix) “마스크 행렬

식(mask matrix)”과 “판정행렬식(decision matrix)”을 구성해야 한다.

그림 3-1은 전도장애에 관한 판정테이블로부터 구성한 마스크 행렬식과 판정행렬식을 나타낸다.

여기서 마스크 행렬식 M은 판정테이블에서 YES나 NO와 같이 응답이 있는 조건을 “1”로써, 응답이 전혀 없는 조건은 “0”으로 각각 나타낸다.

M	D
1 1 1 1 1 1 1 1	1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1	0 1 0 1 0 1 0 1
1 1 1 1 1 1 1 1	1 1 1 1 1 1 0 0
1 1 1 0 1 0 0 0	1 0 0 0 0 0 0 0
0 1 1 1 0 1 0 0	0 1 0 0 0 0 0 0
0 0 1 1 1 0 0 0	0 0 1 0 0 0 0 0
0 0 0 1 0 1 0 0	0 0 0 1 0 0 0 0

그림 3-1. (a) 마스크 행렬식 M  
(b) 판정 행렬식 D

Fig. 3-1. (a) The mask matrix M.  
(b) The decision matrix D.

한편 판정행렬식 D는 응답이 YES인 경우에 “1”로 다른 모든 경우는 “0”으로 나타내서 행렬식을 구성한다.

두번째 단계는 환자의 ECG와 일치하는 항목에 대하여 판정테이블의 각 조건을 검토하여 환자의 데이터 벡터를 만들어야 한다. 이 시험의 결과는 2진벡터로 나타난다. 만일 주어진 조건이 TRUE로 판정되면 1이 기록되고 반대이면 0으로 된다. 여기서 벡터요소의 수는 판정테이블이 주어진 조건의 수와 같아야 한다. 그러므로 만일 전도장애 테이블에서 제1, 3 및 제6번째의 조건이 환자의 데이터벡터에 의하여 만족되고 나머지는 모두 만족되지 않는다면 데이터벡터는 10100100이 된다. 그림 3-2는 전도장애 테이블을 위한 환자의 데이터벡터를 발생시키기 위한 유통도이다. 환자의 데이터벡터는 초기에는 00000000으로 만든다. 그 후에 프로그램상에서 주어진 조건이 만족되면 그 행을 1로 한다. 환자의 데이터벡터는 마스크 행렬식을 검색하는데 사용된다. 마스크 행렬식의 검색은 마스크행렬식의 모든 열과 환자의 데이터벡터 사이의 논리적(logical multiplication)으로 이루어진다. 이 과정을 통하여 주어진 규정에 적합하지 않은 조건들의 결과를 찾아낸다. 논리적으로 의해 산출된 각 벡터는 판정행렬식의 각 열과 일치되는지를 차례로 검색하게 된다.

컴퓨터 프로그램에서 등가연산은 EXCLUSIVE OR 연산으로 바꾸어진다. EXCLUSIVE OR 연산의 결과는 두 벡터의 요소가 같을 때만 “0”이 된다.

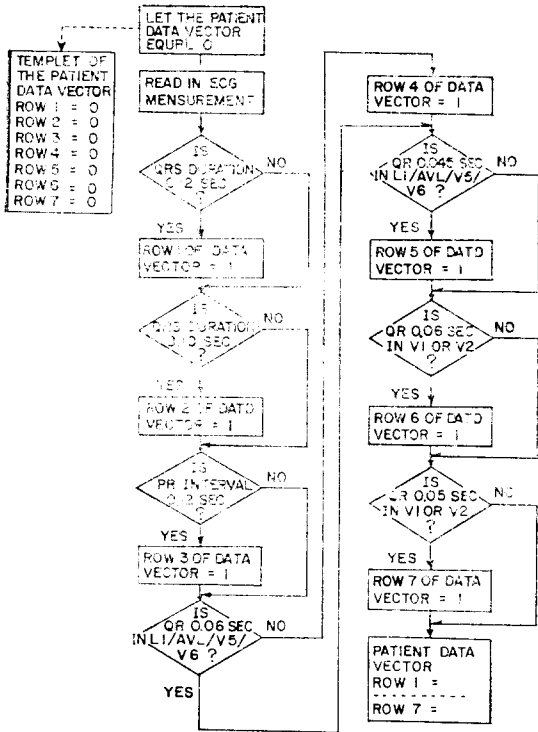


그림 3-2. 심장 전도장애 테이블에 대한 환자의 데이터벡터 발생유통도.  
 Fig. 3-2. Flow chart for generating the patient data vector for the conduction defects table of the heart.

그림 3-3은 전도장애테이블의 첫번째와 두번째 규정에 대한 검색동작을 보여준다. 판정규정 NO.1에 대한 마스크행렬식과 환자의 데이터벡터의 논리적인 판정행렬식의 첫번째 판정규정과 일치되지 않는 벡터를 찾아낸다. 그러나 마스크행렬식의 제2열과 환자의 데이터벡터의 논리적인 판정행렬식의 제2열과 일치되는 벡터를 발생시킨다. 그러므로 두번째 판정규정이 만족된다. 만일 일치되는 판정규정이 없을 경우에는 ELSE 규정이 수행된다. 그림 3-4는 판정테이블의 규정시험을 위한 유통도이다.

3-2. 전도장애 테이블의 FORTRAN 프로그램

ECG 데이터의 진단평가를 종합적으로 제어하는 주 프로그램과 결합되는 전도장애 테이블에 대한 FORTRAN 부프로그램을 구성해 보고자 한다. 이 프로그램은 환자의 데이터벡터 PV(1)을 0으로 놓는 과정부터 시작된다. 즉 이것은 환자의 데이터벡터의 각 요소에 논리치 False을 지정함으로써 이루어진다. 이 환자 데이터벡터의 요소의 수는 전도장애 테이블의 조건수와 같아야 한다.

환자의 ECG에 일치되는 규정항목에 대한 판정테이블에 주어진 조건의 시험은 "산술 IF 명령문"의 이용으로 이루어진다. 이들 명령문의 출력은 세 가지 가능한 수준의 하나로 된다. 즉 세 가지 가능한 수만 만일 ECG 데이터의 내용이 조건으로 주어진 값 이하이면 부(-)의 수로, 같으면 0으로, 그리고 조건값보다 크

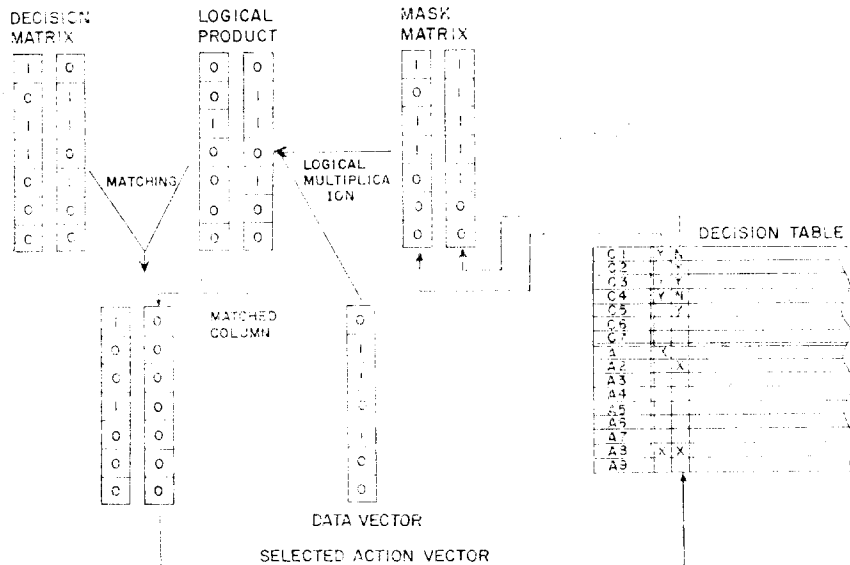


그림 3-3. 환자 데이터벡터에 의한 판정테이블의 검색 동작의 개략도.  
 Fig. 3-3. Schematic diagram illustrating the scanning operation of a decision table by the patient data vector.

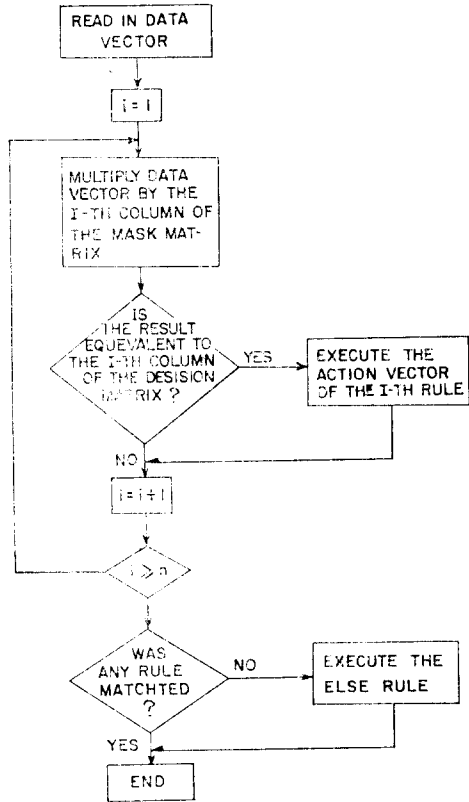


그림 3-4. 판정테이블에서 시험규정에 대한 유통도  
Fig. 3-4. Flow chart for testing rules in a decision table.

면 양의 수로 되는 것을 말한다.

예를 들면

```

30 | IF (QRS DUR—120) 30, 70, 70
    | IF (QRS DUR—100) 40, 80, 80
70 | PV(1) = TRUE.의 의미는
  
```

“만일 QRS 주기가 0.12sec 보다 작으면 컴퓨터는 명령번호 30으로 지시하여 QRS 주기가 0.10sec 보다 큰지 혹은 같은지를 확인시킨다. 또한 만일 QRS 주기가 첫번째 조건의 만족을 의미하는 0.12sec 보다 크거나 같으면 컴퓨터는 환자의 메타벡터의 첫번째 요소의 값을 TRUE.로 지정하는 명령문 번호 70번으로 진행된다. 그러나 어떤 조건들은 몇개의 유도전극으로 주어진 ECG 데이터 내용의 시험을 필요로 하는데 이것을 다중유도조건이라 부른다. 이 조건들이 환자의 메타벡터와 비교되어 그 결과가 서로 일치될 때는 언제나 1이고 그렇지 않으면 0인 2진벡터로 처리된다.

예를들면 lead L<sub>1</sub>, L<sub>2</sub>, L<sub>3</sub>, AVR, AVL, 및 AVF로 측정된 QRS 주기가 전도장에 테이블의 첫번째 조건에

대해 시험되면 이것은 6개의 2진요소 벡터가 된다. 만일 이 벡터가 적어도 2개의 1인 요소를 갖는다면 첫번째 조건은 만족된 것으로 간주된다.

ECG의 유도방법을 간단히 하기 위해 다음과 같이 1(L<sub>1</sub>), 2(L<sub>2</sub>), 3(L<sub>3</sub>), 4(AVR), 5(AVL), 6(AVF), 7(V<sub>1</sub>), 8(V<sub>2</sub>), 9(V<sub>3</sub>), 10(V<sub>4</sub>), 11(V<sub>5</sub>) 및 12(V<sub>6</sub>)인 기호를 사용한다.

이상의 내용을 마이크로 컴퓨터인 PDP11/03를 이용하여 부록에 나타낸 바와 같은 ECG 자동진단을 위한 컴퓨터 프로그램을 개발하였다.

#### 4. 결과 및 고찰

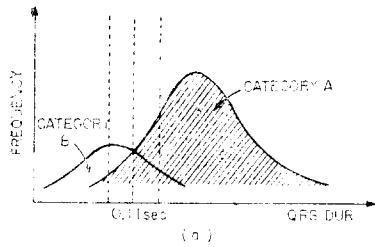
컴퓨터에 의한 의료진단을 위하여 판정테이블 기법을 이용하는 것은 많은 중요한 장점이 있다. 그 중에서도 특히 중요한 것은 의료진단에서 발생하는 복잡한 문제들을 아주 간단한 형태로 나타낼 수 있다는 점이다. 그리고 이 방법은 의사들이나 프로그래머들이 몇개의 작은 테이블을 상호 연결함으로써 복잡한 결정을 내려가는 과정을 단순하게 분할할 수 있다. 판정테이블의 이용도는 이들을 컴퓨터언어로 쉽게 변형시킬 수 있기 때문에 더욱 손쉽게 확장될 수 있다.

한편 판정테이블 기법을 이용하는 진단방법에 주의해야 될 점이 있다. 즉, 임상적 징후는 어떤 특수한 범주에 나타나거나 혹은 갑자기 사라져야 하기 때문이며 대부분의 경우에 그러한 판정특성은 별로 가치가 없다. 왜냐하면 임상적 징후는 상호 벡터적이지 아니라 오히려 상당히 중복되기 때문이다.

그래서 그러한 경우에 판정테이블은 통계적 근거에 의해서 구성되어야 한다.

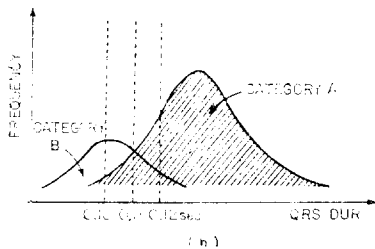
그림 4-1(a)는 의료진단에 대한 결정론적인 접근을 묘사하고 있다. 두 가지의 ECG 진단 범주 A, B를 QRS 주기의 주파수 분포에 의하여 구성하였고 그들 사이의 경계선은 0.11sec로 고정되었다. 그래서 QRS 주기가 0.11sec 보다 작은 경우에는 B에 속하는 것으로 되고 0.11sec 보다 클 경우에는 A에 속하는 것으로 진단된다. 그러나 B범주에 있는 ECG의 약 25%가 A범주의 ECG로 진단될 수 있고, 동시에 A범주에 속하는 ECG의 약 10%가 B범주의 ECG로 진단될 수 있기 때문에 그러한 진단이 부정확하리라는 것은 분명하다.

그래서 그림 4-1(b)는 의료진단에 대한 확률적 접근이 판정테이블로 어떻게 표현될 수 있는가를 보여주며 여기서는 한개의 경계선 대신에 두 가지의 진단범주 사이에 중복의 다른 척도를 나타내는 세 개의 범주로 나타낸다.



Deterministic type of the decision table

QRS < C1/1 SEC	Y	N		
QRS > C1/1 SEC	N	Y		
CATEGORY B			X	
CATEGORY A				X



Probabilistic type of the decision table

QRS < C1/1 SEC	Y	N	N	
QRS < C2/1 SEC	Y	N	N	
QRS < C3/1 SEC	Y	N		
QRS > C3/1 SEC			Y	
CATEGORY B		X		
CONSISTENT CATEGORY A			X	
INCONSISTENT CATEGORY A			X	
CONSISTENT A WITH CATEGORY A				X

그림 4-1. 판정테이블의 두 형태  
 (a) 판정테이블의 결정론적 형태  
 (b) 판정테이블의 확률론적 형태

Fig. 4-1. Two types of decision table.  
 (a) Deterministic type of the decision table.  
 (b) Probabilistic type of the decision table.

5. 결 론

참 고 문 헌

본 연구는 판정테이블 기법에 의한 ECG의 자동진단용 컴퓨터 프로그램의 개발에 관한 연구로 그 얻어진 결과를 요약하면 다음과 같다.

1. 판정테이블 기법은 의료진단에서 병과 증상사이의 많은 복잡한 문제들을 대단히 간단한 형태로 표현할 수 있다.

2. 판정테이블의 형태는 대단히 간단하여 의사들이 사용방법에 관한 전문적인 훈련을 받지 않더라도 또는 실제적으로 컴퓨터나 프로그래밍에 관한 지식이 없어도 사용할 수 있다.

3. 의사나 혹은 프로그래머들이 판정해야 할 복잡한 과정을 상호 관련되는 몇개의 테이블로 분할하여 간단한 컴퓨터 언어로 변환할 수 있다.

4. 본 프로그램은 마이크로 컴퓨터에 응용하면 ECG의 진단을 real-time으로 할 수 있는 간단한 휴대용 ECG 진단 시스템 개발이 가능하다.

5. 또한 다른 종류의 병을 진단하는데 있어서도 본 논문의 프로그램이 쉽게 확장될 수 있다.

이상의 결론으로 미루어보아 본 논문은 임상진단 및 진료의 자동화를 위한 컴퓨터 시뮬레이션의 기초적인 자료가 될 것으로 기대된다.

- 1) Homer R. Warner, "A Mathematical Approach to Medical Diagnosis," J.A.M.A., July 22, Vol. 177, No. 3, pp.177~183, 1961.
- 2) John E. Overall, "Conditional Probability Program for Diagnosis of Thyroid Function", J.A.M.A., Feb. 2, Vol.183, No. 5, pp.307~313 1963.
- 3) O. Rompelman, A.J.R.M. Coenen, "Measurement of Heart Rate Variability: Part I-Comparative Study of Heart Rate Variability Analysis Methods, Med. Biol. Eng. Computer, Vol.15, pp.233~239, 1977.
- 4) J.B.I.M. Sniijders, "A Microcomputer System for the Real Time Assessment of Heart Rate Variability Spectra," (in Dutch), M.S. Thesis., Dept. of Electrical Eng., Delft Univ. Technology, Delft, The Netherlands, 1981.
- 5) C.J. Van Spronsen and O. Rompelman, "Personal Computers in Cardiology," in Microprocessor System, M. Sami, L. Thompson, The Netherlands 1981.
- 6) Otto Rompelman and J.B. Shijders, "The Measurement of Heart Rate Variability Spectra with the Help of a Personal Computer," IEEE Transaction on Biomedical Engineering, Vol BME 29, No. 7, July, pp.503~510, 1982.

부 록

심전도의 자동진단을 위한 프로그램

```

.R FORTRAN
:AAA,IT:=AAA,BBB
FORTRAN IV          V01C-03A                      PAGE 001

0001      CALL ASSIGN(5,'AAA.DAT',7)
0002      LOGICAL A(7),B(7),C(7)
          *,D(7,8),M(7,8),L(7,8),E(7,8),MTA1(7,4),DTA1(7,4),
          *QL80,L1(7,4),E1(7,4),MTA2(7,5),DTA2(7,5),L2(7,5),E2(7,5)
          *,WFW,RBBB,LBBB
0003      REAL LAD
0004      READ(5,79)((M(I,J),J=1,8),I=1,7)
0005      READ(5,89)((D(I,J),J=1,8),I=1,7)
0006      READ(5,1)((MTA1(I,J),J=1,4),I=1,7)
0007      READ(5,2)((DTA1(I,J),J=1,4),I=1,7)
0008      READ(5,3)((MTA2(I,J),J=1,5),I=1,7)
0009      READ(5,4)((DTA2(I,J),J=1,5),I=1,7)
0010      READ(5,7)(A(I),I=1,7)
0011      READ(5,8)(B(I),I=1,7)
0012      READ(5,9)(C(I),I=1,7)
0013      79  FORMAT(8L3)
0014      89  FORMAT(8L3)
0015      1   FORMAT(4L3)
0016      2   FORMAT(4L3)
0017      3   FORMAT(5L3)
0018      4   FORMAT(5L3)
0019      7   FORMAT(7L3)
0020      8   FORMAT(7L3)
0021      9   FORMAT(7L3)
0022      COMMON AGE,RDV1,RSV1,RAD,LAD,RSV5,RSV6,SAV1,RA1,RAVL
0023      READ(5,20)WFW,RBBB,AGE,RDV1,QL80,RSV1,LBBB,RNV1,RAD,LAD,
          *RSV5,RSV6,RAV1,RAA1,RAV5,RAV6,SAV1,RA1,RAVL
0024      READ(5,100)QRS,PR,QRA,QRL,QRV
0025      20  FORMAT(2L3,2F6.3,L3,F6.3,L3,3F6.3/9F6.3)
0026      100 FORMAT(5F7.3)
0027      IF(QRS.GE.0.12) A(1)=.TRUE.
0029      IF(QRS.GE.0.10) A(2)=.TRUE.
0031      IF(PR.GE.0.12) A(3)=.TRUE.
0033      IF(QRA.GE.0.06) A(4)=.TRUE.
0035      IF(QRL.GE.0.045) A(5)=.TRUE.
0037      IF(QRV.GE.0.06) A(6)=.TRUE.
0039      IF(QRV.GE.0.05) A(7)=.TRUE.
0041      WRITE(6,45)(A(I),I=1,7)
0042      45  FORMAT(1X,7L3)
0043      J=1
0044      50  DO 21 I=1,7
0045          L(I,J)=A(I).AND.M(I,J)
0046          E(I,J)=((.NOT.D(I,J)).OR.(.NOT.L(I,J))).AND.
          *(D(I,J).OR.L(I,J))
    
```

```

0047      IF(E(I,J).NE.(.FALSE.)) GO TO 30
0049  21    CONTINUE
0050      WRITE(6,99) J
0051  99    FORMAT(10X,'MACHED RULE= ',I2)
0052      GO TO 98
0053  30    J=J+1
0054      IF(J.LE.8) GO TO 50
0056  98    IF(WPW.EQ.(.TRUE.)) B(1)=.TRUE.
0058      IF(RBBB.EQ.(.TRUE.)) B(2)=.TRUE.
0060      IF(AGE.GE.40) B(3)=.TRUE.
FORTRAN IV      U01C-03A
0062      IF(RDV1.GE.40) B(4)=.TRUE.
0064      IF(QLB0.EQ.(.TRUE.)) B(5)=.TRUE.
0066      IF(RSV1.GE.1.0) B(6)=.TRUE.
0068      IF(LBBB.EQ.(.TRUE.)) B(7)=.TRUE.
0070      J=1
0071  44    DO 29 I=1,7
0072      L1(I,J)=B(I).AND.MTA1(I,J)
0073      E1(I,J)=((.NOT.DTA1(I,J)).OR.(.NOT.L1(I,J))).AND.
*      (DTA1(I,J).OR.L1(I,J))
0074      IF(E1(I,J).NE.(.FALSE.)) GO TO 51
0076  29    CONTINUE
0077      IF(J.EQ.1) GO TO 52
0079      GO TO 666
0080      WRITE(6,70)
0081  70    FORMAT(10X,'RAV OR POSTERIOR IN FARCTION')
0082      GO TO 666
0083  51    J=J+1
0084      IF(J.LE.4) GO TO 44
0086  666   IF(AGE.GE.16) C(1)=.TRUE.
0088      IF(RNV1.GE.1.0) C(2)=.TRUE.
0090      IF(RSV1.GE.1.0) C(3)=.TRUE.
0092      IF(RAD.GE.110) C(4)=.TRUE.
0094      IF(LAD.GE.-31) C(5)=.TRUE.
0096      IF(RSV5.GE.1.0) C(6)=.TRUE.
0098      IF(RSV6.GE.1.0)C(7)=.TRUE.
0100      J=1
0101  49    DO 324 I=1,7
0102      L2(I,J)=C(I).AND.MTA2(I,J)
0103      E2(I,J)=((.NOT.DTA2(I,J)).OR.(.NOT.L2(I,J))).AND.
*      (DTA2(I,J).OR.L2(I,J))
0104      IF(E2(I,J).NE.(.FALSE.)) GO TO 54
0106  324   CONTINUE
0107      IF(J.LE.5) GO TO 325
0109      GO TO 667
0110  325   WRITE(6,376)
0111  376   FORMAT(10X,'RIGHT VETRICULAR ENLARGEMENT')
0112      GO TO 667
0113  54    J=J+1
0114      IF(J.LE.5) GO TO 49
0116  667   CALL ABS
0117      STOP
0118      END

```

PAGE 002



```

0001      SUBROUTINE ABS
0002      LOGICAL DX(7),EX(4),F(5),G(6),MTA3(7,5),DTA3(7,5)
          *,L3(7,5),E3(7,5),MTA4(4,3),DTA4(4,3),
          *L4(4,3),E4(4,3),MTA5(5,3),DTA5(5,3),L5(5,3),E5(5,3),
          *MTA6(6,5),DTA6(6,5),L6(6,5),E6(6,5)
0003      COMMON AGE,RDV1,RSV1,RNV1,RAD,LAD,RSV5,RSV6,
          *SAV1,RA1,RAVL
0004      READ(5,140)(DX(I),I=1,7)
0005      READ(5,141)(EX(I),I=1,4)
0006      READ(5,142)(F(I),I=1,5)
0007      READ(5,143)(G(I),I=1,6)
0008      READ(5,144)((MTA3(I,J),J=1,5),I=1,7)
0009      READ(5,145)((DTA3(I,J),J=1,5),I=1,7)
0010      READ(5,146)((MTA4(I,J),J=1,3),I=1,4)
0011      READ(5,147)((DTA4(I,J),J=1,3),I=1,4)
0012      READ(5,148)((MTA5(I,J),J=1,3),I=1,5)
0013      READ(5,149)((DTA5(I,J),J=1,3),I=1,5)
0014      READ(5,101)((MTA6(I,J),J=1,5),I=1,6)
0015      READ(5,102)((DTA6(I,J),J=1,5),I=1,6)
0016      140  FORMAT(7L3)
0017      141  FORMAT(4L3)
0018      142  FORMAT(5L3)
0019      143  FORMAT(6L3)
0020      144  FORMAT(5L3)
0021      145  FORMAT(5L3)
0022      146  FORMAT(3L3)
0023      147  FORMAT(3L3)
0024      148  FORMAT(3L3)
0025      149  FORMAT(3L3)
0026      101  FORMAT(5L3)
0027      102  FORMAT(5L3)
0028      IF(AGE.GE.30.0) DX(1)=.TRUE.
0029      IF(RAV1.GE.0.7) DX(2)=.TRUE.
0030      IF(RAA1.GE.0.7) DX(3)=.TRUE.
0031      IF(RAD.GE.90.0) DX(4)=.TRUE.
0032      IF(RAV5.GE.2.6) DX(5)=.TRUE.
0033      IF(RAV6.GE.2.6) DX(6)=.TRUE.
0034      IF(SAV1.GE.0.5) DX(7)=.TRUE.
0035      J=1
0036      201  DO 329 I=1,7
          L3(I,J)=DX(I).AND.MTA3(I,J)
          E3(I,J)=((.NOT.DTA3(I,J)).OR.(.NOT.L3(I,J))).AND.(DTA3(I
          *,J).OR.L3(I,J))
0037      IF(E3(I,J).NE.(.FALSE.)) GO TO 401
0038      329  CONTINUE
0039      IF(J.LE.3) GO TO 402
0040      IF(J.GE.4) GO TO 403
0041      402  WRITE(6,303)
          GO TO 668
0042      403  WRITE(6,304)
0043      303  FORMAT(10X,'BIVENTRICULAR ENLARGEMENT')
0044      304  FORMAT(10X,'RIGHT VETRICULAR ENLARGEMENT')

```

```

0058 401 J=J+1
0059 IF(J.LE.5) GO TO 201
FORTRAN IV V01C-03A

0061 668 IF(RA1.GE.3.3) EX(1)=.TRUE.
0063 IF(RAVL.GE.1.1) EX(2)=.TRUE.
0065 IF(AGE.GE.40.0) EX(3)=.TRUE.
0067 SR=SAV1+RAV6
0068 IF(SR.GE.3.5) EX(4)=.TRUE.
0070 J=1
0071 169 DO 166 I=1,4
0072 L4(I,J)=EX(I).AND.MTA4(I,J)
0073 E4(I,J)=((.NOT.DTA4(I,J)).OR.(.NOT.L4(I,J))).AND.
*(DTA4(I,J).OR.L4(I,J))
0074 IF(E4(I,J).NE.(.FALSE.)) GO TO 167
0076 166 CONTINUE
0077 WRITE(6,168)
0078 168 FORMAT(10X,'LEFT VENTRICULAR ENLARGEMENT')
0079 167 J=J+1
0080 IF(J.LE.3) GO TO 169
0082 IF(RAV1.GE.1.3) F(1)=.TRUE.
0084 IF(RAVL.GE.1.1) F(2)=.TRUE.
0086 IF(AGE.GE.40) F(3)=.TRUE.
0088 VS1=SAV1+RAV1
0089 VS2=SAV1+RAV5
0090 IF(VS1.GE.3.5) F(4)=.TRUE.
0092 IF(VS2.GE.3.5) F(5)=.TRUE.
0094 J=1
0095 804 DO 801 I=1,5
0096 L5(I,J)=F(I).AND.MTA5(I,J)
0097 E5(I,J)=((.NOT.DTA5(I,J)).OR.(.NOT.L5(I,J))).AND.
*(DTA5(I,J).OR.L5(I,J))
0098 IF(E5(I,J).NE.(.FALSE.)) GO TO 802
0100 801 CONTINUE
0101 WRITE(6,803)
0102 803 FORMAT(10X,'INCOMPLETE LBBB WITH LVE')
0103 GO TO 1000
0104 802 J=J+1
0105 IF(J.LE.3) GO TO 804
0107 IF(AGE.GE.30.) G(1)=.TRUE.
0109 IF(SAV1.GE.2.4) G(2)=.TRUE.
0111 IF(RAV5.GE.2.6) G(3)=.TRUE.
0113 IF(RAV6.GE.2.2) G(4)=.TRUE.
0115 VV5=SAV1+RAV5
0116 VV6=SAV1+RAV6
0117 IF(VV5.GE.4.0) G(5)=.TRUE.
0119 IF(VV6.GE.4.0) G(6)=.TRUE.
0121 J=1
0122 441 DO 177 I=1,4
0123 L6(I,J)=G(I).AND.MTA6(I,J)
0124 E6(I,J)=((.NOT.DTA6(I,J)).OR.(.NOT.L6(I,J))).AND.
*(DTA6(I,J).OR.L6(I,J))
0125 IF(E6(I,J).NE.(.FALSE.)) GO TO 178
0127 177 CONTINUE

```

PAGE 002

—안재봉 외 : ECG의 자동진단을 위한 컴퓨터 알고리즘—

```
0128      WRITE(6,179)
0129 179  FORMAT(10X,'LEFT VETRICULAR ENLARGEMENT')
0130      GO TO 1000
0131 178  J=J+1
FORTRAN IV      V01C-03A
```

PAGE 003

```
0132      IF(J.LE.5) GO TO 441
0134      WRITE(6,33)
0135 33  FORMAT(10X,'THIS IS THE ELSE RULE')
0136 1000 RETURN
0137      END
```

RUN AAA

```
F T T F T F F
MACHED RULE= 2
RAV OR POSTERIOR IN FARCTION
RIGHT VETRICULAR ENLARGEMENT
LEFT VENTRICULAR ENLARGEMENT
```

STOP --