SYNDROMES FOR MEDICAL DIAGNOSIS: A SEARCH METHOD IN SAS LANGUAGE

Victor Kamensky, Albert Einstein College of Medicine
Ilya Muchnik, RUTCOR, Rutgers University
Leonyd Yamolinsky, New Hope Guild Centers

ABSTRACT

Symptom is an indicator of a health disorder. Syndrome is a combination of symptoms which can occur on sick patients only. Macro THREE finds all syndromes of 1, 2 and 3 symptoms. An example of syndromes on Beck Inventory (Beck & Steer, 1987) is described.

SYMPTOMS, SYNDROMES AND MEDICAL DIAGNOSIS

Medical diagnosis starts from symptoms. Symptom is mostly an indicator of a health disorder, for example, high temperature, high blood pressure etc. A group of symptoms that occur together is called a syndrome. Stedman's Medical Dictionary notes that syndrome is an aggregate of signs or symptoms associated with any morbid process and constituting together the picture of a disease. Syndrome literally means “running together.” A syndrome includes symptoms not statistically, but in connection with the logic of disease. Each disease has a specific set of syndromes.

Examples:

- Depressive Syndrome (depressive triad) - low mood; intellectual inhibition; motor inhibition.
- Korsakoff (amnestic) Syndrome - confusion and severe impairment of memory, especially for recent events, for which the patient compensates by confabulation (impaired memory; confusion, disorientation; confabulation).

Diagnosis is the determination of the individual nature of a disease based on a set of syndromes. Each individual case does not include all syndromes. Useful syndromes are usually short - not more than 3 (as in macro THREE below) or 4 symptoms.

WHAT IS A SYNDROME? A SAS DESCRIPTION

All items describing each patient are kept as a Boolean character variable X, i.e. each character of the variable X is either 0 or 1. We will call it Boolean vector.

Patients are divided into two classes - sick ("T") and healthy ("F").

A subvector of a Boolean vector X is the character variable for which some elements are equal to the elements of X, and others are substituted by '·'. Length of a subvector is number of elements not equal '·'.

Example 1:

```
Boolean vector X : 100000110001
A subvector of X (length 3): --0--1--0--
```

DEFINITION: syndrome is a subvector which can occur only on one of two classes.

In terms of Logical Analysis of Data-LAD (see CRAIK, HAMMER and IBARAKI, 1988; BOROS, HAMMER, KAMENSKY, KOGAN and MUCHNIK, 1995) syndromes are implicants.

Syndrome is called pure if it does not have shorter syndromes as subvectors.

Syndromes of length 1 will be called SYMPTOMS.

Syndromes of length 2 will be called PAIRS.

Syndromes of length 3 will be called TRIPLETS.

Example 2:

```
CLASS T
1110
0111

CLASS F
1100
1010
0110

CLASS T - PURE SYNDROMES
---1 SYMPTOM
111- TRIPLET

CLASS F - PURE SYNDROMES
--0- SYMPTOM
-0-- SYMPTOM
0--0 PAIR
```

Next section describes the SAS macro designed to find all symptoms, pairs and triplets.
THREE - A SAS MACRO TO FIND SYNDROMES

A simple, but not a very good idea is the following: generate for each class all existing combinations of 1, 2 and 3 variables and keep those which occur on one of the two classes only. The generation of all combinations can be done by the following macro (&LN is number of element in vector X):

%MACRO GENTHREE(DS,THREE);
DATA &THREE(KEEP=X); SET &DS(KEEP=X);
LENGTH XX $ &LN; XX=X;
DO I3 = 1 TO &LN;
DO I2 = I3 TO &LN;
DO I1 = I2 TO &LN;
IF I3=I1 AND I2=I1 THEN 
  SUBSTR(X,I,1)='-';
END;
OUTPUT;
X=XX;
END;
END:
END:
RUN;
PROC SORT SORTSIZE=15000 DATA=&THREE: BY X:RUN;
DATA &THREE; SET &THREE; BY X; IF LAST.X;RUN;
%MEND GENTHREE;

All syndromes can be found by the macro THREE_S.
&DT-sick ("*"") patients, &DF-healthy ("#") patients.
&T_THREE-syndromes for class T.
&F_THREE-syndromes for class F.

%MACRO THREE_S(DT,DF,T_THREE,F_THREE);
%GENTHREE(DT,T_THREE);
%GENTHREE(DF,F_THREE);
DATA &T_THREE &F_THREE ;
  MERGE &T_THREE(IN=INT) &F_THREE(IN=INF);BY X;
  IF '('INT AND INF); IF INT THEN OUTPUT &T_THREE;
  IF INF THEN OUTPUT &F_THREE;
RUN;
%MEND THREE_S;

This macro works fine for short vectors X (&LN <= 10). But for long vectors it needs a lot of disk space to write all combinations and a lot of time to sort them. Another simple solution - do the same for each combination of 13-12-11 separately, i.e. use these variables as macro variables and perform the whole process inside 3 macro loops - takes a lot of time.

The result of macro THREE_S are all syndromes, not only pure.

Macro THREE (see below) presents a better solution and returns pure syndromes only.

It is based on the list of all possible subvectors for 3 variables. This list is shown in the array THR_AR (macro TAR3S).

Array THR_AR has 26 elements. These elements are all possible symptoms, syndromes and triples on 3 variables.

Arrays ARj (j=0-7), used by macro S3, keep numbers of elements in THR_AR which correspond to all subvectors of triple j.

Arrays AI (i=1-18), used by macro REDUCE3, keep numbers of elements in THR_AR which have THR_AR(i) as it's subvector. For example, THR_AR(18) is a subvector of both THR_AR(25) and THR_AR(26).

The main part of macro THREE is the data step in macro THREE_P. Let's describe it's steps (letters A-J are shown on the macro THREE_P below).

Parameters &DT, &T_THREE and &F_THREE have the same meaning as above.

A. Arrays THR_T(class T) and THR_F(class F) keep all existing combinations of values for three variables 13, 12, 11. These combinations are coded as octal numbers from 0 to 7.

B. Arrays SND_T (class T) and SND_F(class F) keep all existing combinations of 1, 2 and 3 variables (from the same 13, 12 and 11 as in A).

C. Loops with 13, 12 and 11 enumerate each subset of 3 different variables from all &IN variables.

D and E. These loops find all existing combinations on variables 13, 12, 11 for classes T and F. It is done by going through each data set until the end or until all 8 possible combinations are found.

For example 2 above results for 13=1 12=2 11=3 are the following:

Step D (CLASS T)
  TRIPLET=7
  TRIPLET=3

ARRAY THR_T(C=7) : 0 0 0 1 0 0 1
S_T=2

Step E (CLASS F)
  TRIPLET=6
  TRIPLET=5
  TRIPLET=3

ARRAY THR_F(C=7) : 0 0 0 1 0 1 0
S_F=3.
F. Macro S3 'unfolds' all existing combinations of three variables into all existing combinations of 1, 2 and 3 variables.

For example 2 above results of step F for 13=1 12=2 11=3 are the following:

\[\text{ARRAY SND}_F{1:26} = \{1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 0 1 1 0 \} \]
\[\text{ARRAY SND}_T{1:26} = \{0 1 0 1 1 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0 0 1 0 0 1\} \]

G. This loop checks, which combinations occur on one of two classes (T and F) only.

For example 2 above results of step G for 13=1 12=2 11=3 are the following:

\[\text{ARRAY SND}_T{1:26} = \{0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1\} \]
\[\text{ARRAY SND}_F{1:26} = \{1 0 1 0 0 0 1 1 0 0 0 1 0 0 0 1 0 0 0 1 1 0 1 0 0\} \]

H. Macro REDUCE3 keeps only pure syndromes.

For example 2 above results of step H for 13=1 12=2 11=3 are the following:

\[\text{ARRAY SND}_T{1:26} = \{0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1\} \]
\[\text{ARRAY SND}_F{1:26} = \{1 0 1 0 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 1 1 0 1 0 0\} \]

I. This loop outputs syndromes into data sets &T_THREE and &F_THREE.

J. All symptoms can be found when 13=1 and 12=2.

All pairs can be found when 13 = 1.

HOW TO SELECT IMPORTANT SYNDROMES

Number of all syndromes is usually big. One of the ways to shorten the list is described in (BOROS, HAMMER, KAMENSKY, KOGAN and MUCHNIK, 1995): minimal cover (macro CDESCRIP).

Another method is based on calculating average Hamming distances from syndromes to all vectors of its own class and another class. Syndromes which are close to its own class and far from another class are selected for further analysis. If the list is still big, it can be shortened by macro CDESCRIP.

EXAMPLE-SYNDROMES BASED ON BECK INVENTORY

Beck Depression Inventory (BDI) was developed by Aaron T. Beck and his associates at the Center for Cognitive Therapy at the University of Pennsylvania (Beck, 1967, 1976; Beck & Beamesderfer, 1974; Beck & Steer, 1987). The Beck Depression Inventory is a reliable mood-measuring device that is used in the presence of depression and precisely rates its severity. It is a simple multiple-choice questionnaire consisting of 21 questions. Each question has four possible answers. These four answers are based on an ordinal scale, with levels changing from 0 to 3. A Total Score is obtained by adding the scores for each of the twenty-one questions. The more severely depressed individuals have the higher Total Score. In contrast, the lower the score, the better the patient feels. The individual Total Score can also be evaluated into four classes ordered along scale.

We used cut point 18 on the BDI Total Score (Score <=18 was considered as non-depressed-Class F, Score > 18 - as depressed-Class T). Certainly, such classification is often not enough for clinical differentiation, but it is very clear and stable for interpretation purposes.

The experimental group has 303 patients selected anonymously from a large database at the New Hope Guild Centers for Emotionally Disturbed Children, Brooklyn, NY. Each patient filled out the BDI. The administration of the BDI was included in the process of psychotherapy of patients with various problems.

Our experiments consisted of two different parts. In the first part we implemented the procedure described in (BOROS, HAMMER, KAMENSKY, KOGAN and MUCHNIK, 1995): divided all data into two samples, constructed a decision-making rule based on one sample and checked it on the other.

Results:

<table>
<thead>
<tr>
<th>DECISION CLASS (F - BDI&lt;=18; T - BDI &gt;= 19)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>F</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

In the second part we made calculation on the whole data set and analyzed good syndromes. Lists of syndromes for both classes are given below.

---

In the second part we made calculation on the whole data set and analyzed good syndromes. Lists of syndromes for both classes are given below.
ACKNOWLEDGEMENTS

The work of one of the authors (Dr Ilya Muchnik) was supported in part by a grant from National Science Foundation NSF-CCR-91-1999 and a grant from AFOSR No F49620-93-1-0041.

REFERENCES


List of syndromes for class T-Depressed patients

<table>
<thead>
<tr>
<th>SYNDROME</th>
<th>N. this patients</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crying &amp; Social Withdrawal &amp; Indecisiveness</td>
<td>48</td>
</tr>
<tr>
<td>Sad Mood &amp; Crying &amp; Fatigability</td>
<td>46</td>
</tr>
<tr>
<td>Sad Mood &amp; Sense of Failure</td>
<td>46</td>
</tr>
<tr>
<td>Sad Mood &amp; Crying &amp; Weight Loss</td>
<td>44</td>
</tr>
<tr>
<td>Pessimism &amp; Sense of Failure &amp; Social Withdrawal</td>
<td>42</td>
</tr>
<tr>
<td>Sad Mood &amp; Pessimism &amp; Loss of Libido</td>
<td>41</td>
</tr>
<tr>
<td>Indecisiveness &amp; Loss of Appetite &amp; Weight Loss</td>
<td>40</td>
</tr>
<tr>
<td>Guilt &amp; Self-dislike &amp; Indecisiveness</td>
<td>40</td>
</tr>
<tr>
<td>Sad Mood &amp; Crying &amp; Irritability</td>
<td>39</td>
</tr>
<tr>
<td>Body Image Change &amp; Fatigability</td>
<td>38</td>
</tr>
<tr>
<td>Pessimism &amp; Self-accusation &amp; Social Withdrawal</td>
<td>38</td>
</tr>
<tr>
<td>Irritability &amp; Work Difficulty</td>
<td>36</td>
</tr>
<tr>
<td>Crying &amp; Work Difficulty &amp; Loss of Appetite</td>
<td>36</td>
</tr>
<tr>
<td>Body Image Change &amp; Weight Loss</td>
<td>35</td>
</tr>
<tr>
<td>Irritability &amp; Weight Loss</td>
<td>32</td>
</tr>
</tbody>
</table>

List of syndromes for class F-Non-Depressed patients

<table>
<thead>
<tr>
<th>SYNDROME</th>
<th>N. this patients</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good Mood &amp; Self-satisfaction &amp; Appetite Unchanged</td>
<td>37</td>
</tr>
<tr>
<td>Good Mood &amp; Sociable &amp; Body Image Unchanged</td>
<td>34</td>
</tr>
<tr>
<td>Optimism &amp; No Crying &amp; Not tired</td>
<td>30</td>
</tr>
<tr>
<td>Optimism &amp; No Crying &amp; Decisiveness</td>
<td>28</td>
</tr>
<tr>
<td>Optimism &amp; Sociable &amp; No Sleep problems</td>
<td>27</td>
</tr>
<tr>
<td>Sociable &amp; Body Image Unchanged &amp; Purposely try to lose Weight</td>
<td>13</td>
</tr>
</tbody>
</table>
%LET L_IMP=3;
%MACRO TAR33;
%* THR_AR=ALL POSSIBLE SYMPTOMS, SYNDROMS AND TRIPLES ON 3 VARIABLES;
ARRAY THR_AR(3**&L_IMP-1) 3 _TEMPORARY_;
"'0-0' '0-1' '1-0' '1-1'
'00' '01' '10' '11'
'00' '01' '10' '11'
'000' '001' '010' '011' '100' '101' '110' '111');
%* IF THE OCTAL NUMBER CORRESPONDING TO A TRIPLE EQUALS j (j=0-7), THEN ARRAY ARj KEEPS NUMBERS OF ALL ITS SUBVECTORS IN THR_AR. ARRAYS AR0-AR7 ARE USED BY MACRO S3;
ARRAY ARO(7) _TEMPORARY_ (1 3 5 7 9 11 15 19);
ARRAY ARO2(7) _TEMPORARY_ (1 4 5 9 11 16 21);
ARRAY ARO5(7) _TEMPORARY_ (1 4 5 9 11 16 21);
ARRAY ARO6(7) _TEMPORARY_ (1 4 6 9 13 17 23);
ARRAY ARO7(7) _TEMPORARY_ (1 4 6 9 13 17 23);
%* EACH OF ARRAYS A1-A18 CORRESPONS TO ONE OF THE FIRST 18 ELEMENTS OF ARRAY THR_AR. IF, FOR EXAMPLE, THR_AR(1) ('0-0') IS A SYMPTOM, ALL COMBINATIONS LISTED IN ARRAY A1 ARE NOT PURE. ARRAYS A1-A18 ARE USED BY MACRO REDUCE3;
ARRAY A13(8) _TEMPORARY_ (7 9 11 13 19 21 23 25);
ARRAY A12(8) _TEMPORARY_ (8 10 12 14 20 22 24 26);
ARRAY A13(8) _TEMPORARY_ (7 8 15 17 19 20 23 24);
ARRAY A14(8) _TEMPORARY_ (9 10 16 18 21 22 25 26);
ARRAY A15(8) _TEMPORARY_ (11 12 15 16 19 20 21 22);
ARRAY A16(8) _TEMPORARY_ (13 14 17 18 23 24 25 26);
ARRAY A17(8) _TEMPORARY_ (19 23);
ARRAY A18(8) _TEMPORARY_ (20 24);
ARRAY A19(2) _TEMPORARY_ (21 25);
ARRAY A10(2) _TEMPORARY_ (22 26);
ARRAY A11(2) _TEMPORARY_ (19 21);
ARRAY A12(2) _TEMPORARY_ (20 22);
ARRAY A13(2) _TEMPORARY_ (23 25);
ARRAY A14(2) _TEMPORARY_ (24 26);
ARRAY A15(2) _TEMPORARY_ (19 20);
ARRAY A16(2) _TEMPORARY_ (21 22);
ARRAY A17(2) _TEMPORARY_ (23 24);
ARRAY A18(2) _TEMPORARY_ (25 26);
%MEND TAR33;

SAS is a registered trademark of SAS Institute Inc. in the USA and other countries. (®) indicates USA registration.
%MACRO THREE_PCDT,DF,T_THREE,F_THREE); %THREE_P.SAS;

DATA &T_THREE(KEEP=X ) &F_THREE (KEEP=X );
LENGTH XX $ &LN ; XX=REPEAT('-',XEVAL(&LN-1));
%TARS3;

%A; ARRAY THR_T(0 : XEVAL(2**&L_IMP-1)); ARRAY THR_F(0 : XEVAL(2**&L_IMP-1));
%B; ARRAY SND_T(0 : XEVAL(3**&L_IMP-1)); ARRAY SND_F(0 : XEVAL(3**&L_IMP-1));

N_T&L_IMP=0; N_F&L_IMP=0;
%C; DO 13 = 1 TO XEVAL(&LN+1-3);
   DO 12 = (13+1) TO XEVAL(&LN+1-2);
   DO 11 = (12+1) TO XEVAL(&LN+1-1):
   DO J=0 TO XEVAL(2**&L_IMP-1): THR_T{J}=0; THR_F{J}=0; END;
   S_T=0; DO J=1 TO N_T UNTIL( S_T=XEVAL(2**&L_IMP));
      SET DT(KEEP=X ) NOBS=N_T POINT=J:
      TRIPLET= 4 * INPUT(SUBSTR(X,13,1),1.) +
               2 * INPUT(SUBSTR(X,12,1),1.) +
               INPUT(SUBSTR(X,11,1),1.);
      IF THR_T{TRIPLET}=0 THEN DO: THR_T{TRIPLET}=1; S_T=S_T+1; END;
      END:
   END:
   S_F=0; DO J=1 TO N_F UNTIL( S_F=XEVAL(2**&L_IMP));
      SET DF(KEEP=X ) NOBS=N_F POINT=J:
      TRIPLET= 4 * INPUT(SUBSTR(X,13,1),1.) +
               2 * INPUT(SUBSTR(X,12,1),1.) +
               INPUT(SUBSTR(X,11,1),1.);
      IF THR_F{TRIPLET}=0 THEN DO: THR_F{TRIPLET}=1; S_F=S_F+1; END;
      END:
   IF (S_T < XEVAL(2**&L_IMP) OR (S_F < XEVAL(2**&L_IMP)) THEN DO ;
      DO J=1 TO XEVAL(3**&L_IMP-1); SND_T(J)=0; SND_F(J)=0; END;
      &%S3(SND_T,THR_T); &%S3(SND_F,THR_F);
   END:
   DO 11 = (12+1) TO XEVAL(3**&L_IMP-1):
   IF SND_F(J)=1 AND SND_T(J)=1 THEN DO: SND_F(J)=0; SND_T(J)=0; END;
   END:
%H; %REDUCE3(SND_T); %REDUCE3(SND_F);
%I; DO 11 = (12+1) TO XEVAL(3**&L_IMP-1):
   IF SND_F(J)=1 OR SND_T(J)=1 THEN DO ;
      IF ( J > XEVAL(3**&L_IMP-1-2**&L_IMP) ) OR
         ( XEVAL(2**&L_IMP) < J <= XEVAL(3**&L_IMP-1-2**&L_IMP) )
            AND 13=1 ) OR
         ( XEVAL(2**&L_IMP) <= J AND 13=1 AND 12=2 ) THEN DO :
            X=XX;
            SUBSTR(X,13,1)=SUBSTR(THR_AR(J),1,1);
            SUBSTR(X,12,1)=SUBSTR(THR_AR(J),2,1);
            SUBSTR(X,11,1)=SUBSTR(THR_AR(J),3,1);
            IF SND_T(J)=1 THEN DO; OUTPUT &T_THREE;
               IF J > XEVAL(3**&L_IMP-1-2**&L_IMP) THEN N_T&L_IMP=N_T&L_IMP+1;
               END;
            IF SND_F(J)=1 THEN DO; OUTPUT &F_THREE;
               IF J > XEVAL(3**&L_IMP-1-2**&L_IMP) THEN N_F&L_IMP=N_F&L_IMP+1;
               END;
            END;
      END;
   END:
%J; STOP; RUN;