Paper 184-2012

Clustering Physicians Based on Professional Proximity Using SAS®

N. Yaraghi*, A. Y. Du*, R. Sharman*, R. Gopal**, R. Ramesh*, R. Singh*, G. Singh*

*SUNY at Buffalo, Buffalo, NY, United States

**University of Connecticut, Storrs, CT, United States

ABSTRACT

We show how SAS can be used to conduct a network analysis of physicians in order to identify the professional proximity of different specialties based on the common patients flow. The professional proximities can be used as a basis for further research in various topics such as adoption and usage patterns of Healthcare Information Exchange systems. We are building this paper on the previous paper by Zheng (2011) .In the current paper, we provide clearer codes and present the results based on real data, and we further expand the previous paper by showing the differences in Multi-Dimensional Scaling (MDS) analysis results when the number of dimensions increase. Moreover, we show how the outputs of MDS analysis can be used to conduct a cluster analysis.

INTRODUCTION

Healthcare has recently attracted so much attention from many researchers in different fields and has become a hot research topic. Due to the availability of rich data sets in healthcare, researchers can have more robust analysis based on secondary data and avoid the biases and reliability problems that may be inherent in survey data. For example, conducting social network analysis is much more reliable if real data of patient flow is used rather than personal judgments of respondents in a network, needless to say that conducting a survey from all the members of a large network is almost impossible.

PATIENT- PHYSICIANS DATA SETS

Two data sets are used in this paper. The first data set consists of 42000 observations of de-identified patient records. This is basically a log file of physicians' access to HIE network and shows who has accessed a specific patient's records. The second data set contains the name and specialty of physicians.

The following tables depict the structure of the two data sets. Note that if a patient has been visited by more than 1 physician, then there is more than one observation in the data set.

Patient ID	Physician
Patient 1	John
Patient 2	John
Patient 3	Smith
Patient 3	Kelly
Patient 3	Rajiv
Patient 4	Rajiv

Table 1. Patient/Physician Data Base

Physician	Specialty
John	Podiatrist
Smith	obstetrics & gynecology
Kelly	internal medicine
Rajiv	emergency medicine

Table 2. Physician Specialty Data Base

Since we are interested in analyzing the proximities between specialties, we shall merge the two data sets so that the specialty of each physician is also added to the first data set.
The following code is used to merge these data sets

```
DATA data_set3;
MERGE
       Date_Set1(IN = BD)
       Data_Set2;
   BY Physician;
   IF BD;
RUN;
```

The result will be as follows

Patient ID	Physician	Specialty
Patient 1	John	Podiatrist
Patient 2	John	Podiatrist
Patient 3	Smith	obstetrics & gynecology
Patient 3	Kelly	internal medicine
Patient 3	Rajiv	emergency medicine
Patient 4	Rajiv	emergency medicine

Table 3. Patient/Physician/Specialty Data Base

The following code, changes the structure of data set 3 so that it can be represented in matrix format on Specialty basis.

```
PROC TABULATE DATA=data_set3 out=data_set4;
      CLASS patientID Speciality;
      TABLE Speciality ALL, patientID*N;
RUN;
```

Specialty	Patient 1	Patient 2	Patient 3	Patient 4
Podiatrist	1	1	.	.
obstetrics gynecology	.	.	1	.
internal medicine	.	.	1	.
emergency medicine	.	.	1	1

Table 4. Patient /Specialty Matrix

MEASURING SIMILARITY BETWEEN SPECIALTIES

Now we need to have the number of common patients between each pair of specialties. We would use PROC SQL to construct the edge-list table of specialties based on data set 3 and data set 4.

```
PROC SQL ;
CREATE TABLE edge_list AS
SELECT tbl1.speciality AS speciality1, tbl2.speciality AS speciality2,
count(*) AS Common_patients
FROM data_set3 AS tbl1, data_set4 tbl2
WHERE tbl1.PatientID = tbl2.PatientID
GROUP BY speciality1, speciality2;
QUIT;
```

The results are presented in the following table, note that the specialties which have no common patients with each other are not present in this data set, moreover, the number of common patients between two similar pairs of specialties, shows the total number of patients that specific specialty has visited. For example, Podiatrists had only 1 patient which is reflected in the first row of the next table as the number of common patients while emergency medicine has visited two patients which is reflected by the pair of (emergency medicine, emergency medicine) in the next table.

To calculate the ratio of common patients, we should calculate the following (Zheng, 2011).

$$Overlap\ Ratio = \frac{2 \times common\ patients}{total\ patients\ of\ speciality\ 1 + total\ patients\ of\ speciality\ 2}$$

Speciality1	Speciality2	Common_patients
Podiatrist	Podiatrist	1
obstetrics gynecology	obstetrics gynecology	1
obstetrics gynecology	internal medicine	1
obstetrics gynecology	emergency medicine	1
internal medicine	internal medicine	1
internal medicine	emergency medicine	1
internal medicine	obstetrics gynecology	1
emergency medicine	emergency medicine	2
emergency medicine	internal medicine	1
emergency medicine	obstetrics gynecology	1

Table 5. Edge List Dataset

The following code, extracts the total number of patients in each specialty and calculates the overlap ratio

```
DATA total;
SET edge_list;
IF Speciality1= Speciality2 THEN
total= Common_patients;
ELSE DELETE;
DROP Common_patients;
SPECIALITY=Speciality1;
DROP Speciality1 Speciality2;
RUN;

PROC SQL;
CREATE TABLE number1 AS
SELECT * FROM edge_list, total
WHERE total.SPECIALITY=edge_list.Speciality1;
QUIT;

PROC SQL;
CREATE TABLE number2 AS
SELECT * FROM edge_list, total
WHERE total.SPECIALITY=edge_list. Speciality2;
QUIT;

DATA number1;
SET number1;
total1= total;
RUN;

DATA number2;
SET number2;
total2= total;
RUN;

DATA ratio;
MERGE number1 number2;
BY Speciality1;
OVERLAP_RATIO=(2* Common_patients)/(total1+total2);
keep Speciality1 Speciality2 Common_patients OVERLAP_RATIO;
RUN;
```

Speciality1	Speciality2	Common_patients	Overlap ratio
Podiatrist	Podiatrist	1	1
obstetrics gynecology	obstetrics gynecology	1	1
obstetrics gynecology	internal medicine	1	1
obstetrics gynecology	emergency medicine	1	0.5
internal medicine	internal medicine	1	1
internal medicine	emergency medicine	1	0.5
internal medicine	obstetrics gynecology	1	1
emergency medicine	emergency medicine	2	1
emergency medicine	internal medicine	1	0.5
emergency medicine	obstetrics gynecology	1	0.5

Table 6. Ratio Dataset

PREPARING DATA FOR MULTI DIMENSIONAL CLUSTERING

The following code creates a matrix of similarities between different specialties.

```
PROC FORMAT;
INVALUE INDEX
"Podiatrist" = 1
"obstetrics  gynecology" = 2
"internal medicine" = 3
"emergency medicine" = 4
Run;

DATA similarity_matrix;
ARRAY D(1: &DimMax.);
DO UNTIL(last. Speciality1);
SET RATIO;
BY Speciality1 Speciality2;
IF first.npi1 THEN CALL missing(of D(*));
D(input(Speciality2,INDEX.)) = OVERLAP_RATIO;
Var = Speciality1;
IF last. Speciality1 THEN OUTPUT;
END;
DROP Common_patients;
RUN;
```

	D1	D2	D3	D4	
D1	1	.	.	.	Podiatrist
D2	.	1	1	0.5	obstetrics gynecology
D3	.	1	1	0.5	internal medicine
D4	.	0.5	0.5	1	emergency medicine

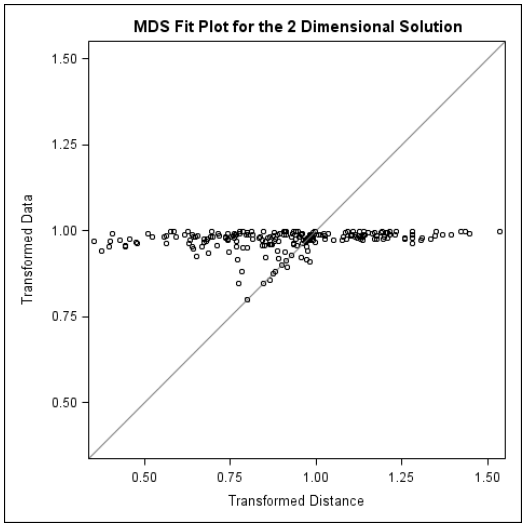
Table 7. Similarity Matrix Dataset

MULTIDIMENSIONAL CLUSTERING ANALYSIS

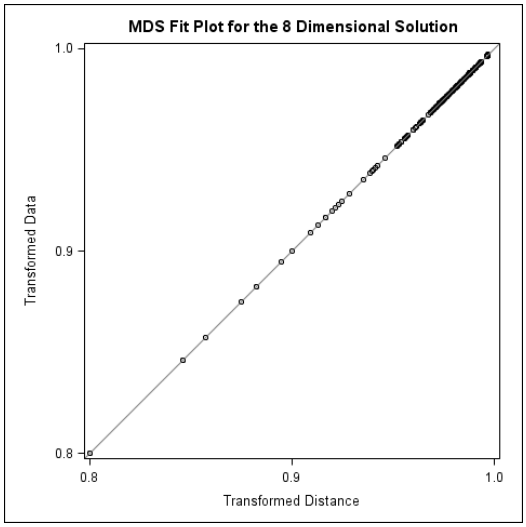
Now we have created a proper data set which can be used as an input for Multi Dimensional Scaling analysis.

```
ODS GRAPHICS ON;
PROC MDS
DATA= similarity_matrix
OUT=MdsTbl
DIMENSION = 2
LEVEL=ratio SIMILAR=1 FIT=1; QUIT;
```

As shown in the code, we have only used 2 dimensions in PRPC MDS, the output 1 shows the fit plot for 2 dimensions.



Output 1. Output from PROC MDS with 2 dimensions



Output 2. Output from PROC MDS with 8 dimensions

The plot does not reflect a good fit and suggests that we may need to increase the dimensions. The output 2 shows the fit with 8 dimensions

Since the transformed distance exactly fits the data, we use the generated 8 dimensions to cluster the specialties.

CLUSTERING SPECIALTIES BASED ON MDS ANALYSIS

Because the variables in the data set do not have equal variance, we must perform some form of scaling or transformation. One method is to standardize the variables to mean zero and variance one (SAS/STAT(R) 9.2 User's Guide).

The following statements perform the ACECLUS transformation by using the SAS data set MdsTbl. The OUT= option creates an output SAS data set called *Ace* to contain the canonical variable scores:

```
PROC ACECLUS DATA= MdsTbl OUT=Ace P=.03 NOPRINT;
VAR DIM1 DIM2 DIM3 DIM4 DIM5 DIM6 DIM7 DIM8;
RUN;

PROC CLUSTER DATA=ACE METHOD=WARD CCC PSEUDO PRINT=15 OUTTREE=tree;
VAR can1 can2 can3 can4 can5 can6 can7 can8;
ID NAME;
RUN;
```

The OUTTREE= statement saves the results to be used for creating a hierarchical representation of the clusters. We can use this data set as an input to PROC TREE procedure.

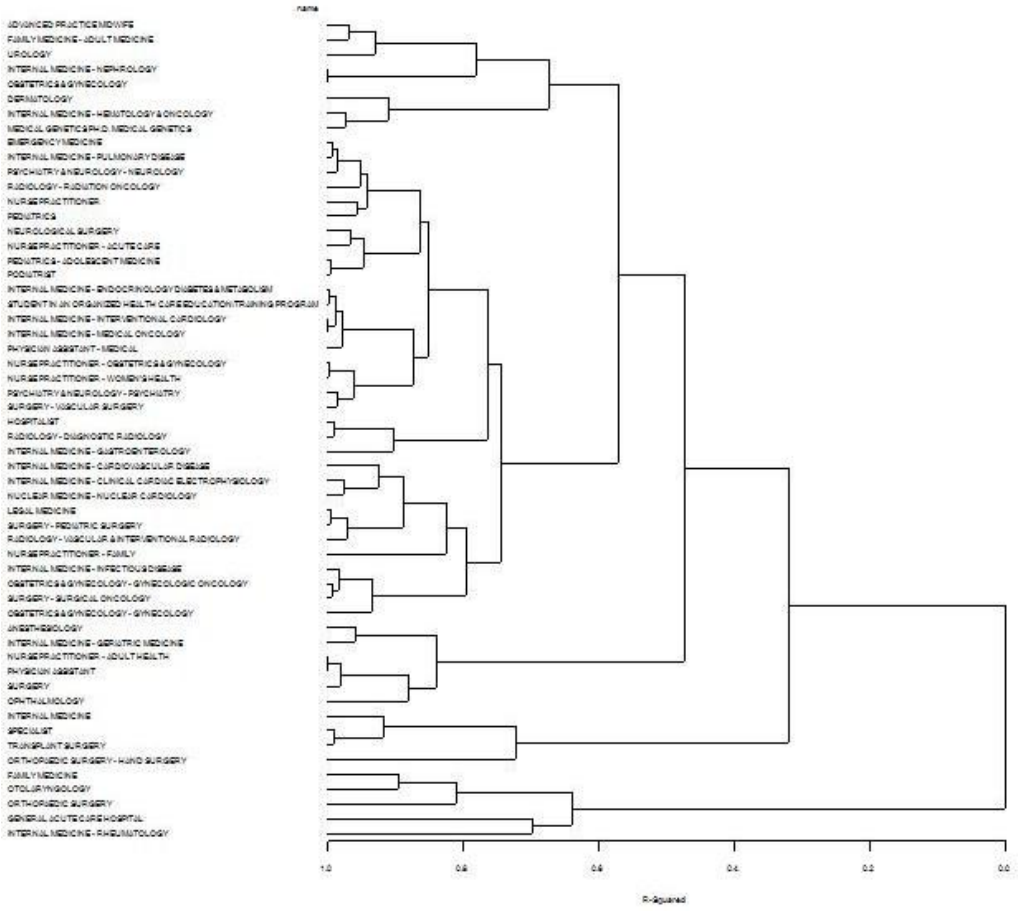
```
GOPTIONS VSIZE=9IN HSIZE=6.4IN HTEXT=.9PCT HTITLE=3PCT;
AXIS1 ORDER=(0 TO 1 BY 0.2);
PROC TREE DATA=tree OUT=new NCLUSTERS=6
      HAXIS=AXIS1 HORIZONTAL;
      HEIGHT _RSQ_;
      ID name;
RUN;

PROC PRINT DATA=new;
RUN;
```

When the number of nodes (in this example specialties) are high, the tree diagram may not be very tangible, the statement OUT= in the tree procedure, prints the clusters ad their members so that it would be easier to distinguish each cluster's members. For example, the members of the cluster number 6 are shown in output 1.

Obs	NAME	CLUSTER	CLUSNAME
1	INTERNAL MEDICINE - NEPHROLOGY	1	CL6
2	OBSTETRICS & GYNECOLOGY	1	CL6
3	INTERNAL MEDICINE - HEMATOLOGY & ONCOLOGY	1	CL6
4	MEDICAL GENETICS PH.D. MEDICAL GENETICS	1	CL6
5	ADVANCED PRACTICE MIDWIFE	1	CL6
6	FAMILY MEDICINE - ADULT MEDICINE	1	CL6
7	UROLOGY	1	CL6

Output 3. Output from PROC TREE statement

The following tree diagram shows the clustering results of 57 different specialties based on their common patients.



Output 4. Graphical Output from PROC TREE statement

RECOMMENDED READING

Zheng,J. Visualizing Healthcare Provider Network using SAS® Tools, PharmaSUG201.
http://www.pharmasug.org/proceedings/2011/HS/PharmaSUG-2011-HS08.pdf

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Niam Yaraghi
Enterprise: SUNY at Buffalo
Address: 304 Jacobs Management Center,
City, State ZIP: Buffalo, NY 14260-4000
Work Phone: (716)645-5256
E-mail: niamyara@buffalo.edu