Paper 259-27

A SAS® Application for Measuring Efficiency and Productivity of Decision Making Units

Ali Emrouznejad

http://www.deazone.com
A_Emrouznejad@hotmail.com

ABSTRACT

In the last two decades many authors have developed techniques to measure the efficiency of Decision Making Units in a way consistent with the underlying economic theory of optimization behavior. The model frequently used is Data Envelopment Analysis (DEA) which initially developed by Charnes, Cooper and Rhodes (1978). Using such as efficiency method over two consecutive periods, one can naturally define productivity of Decision Making Units.

The aim of this paper is to present the capability of SAS® and SAS/OR® to implement such models. Therefor, the paper provides SAS code for various productivity measures using optimization procedures such as PROC LP.

The code can be used to calculate non-parametric Malmquist Index as a base measure for productivity.

KEYWORDS

SAS, DEA, Malmquist Index, Efficiency and Productivity

INTRODUCTION

Data Envelopment Analysis (DEA) is a linear programming method for assessing the efficiency and productivity of Decision Making Units (see Cooper, Seiford and Tone (1999) for details). DEA continues to grow in importance as managerial tools become more reliable and handle performance measurement of organizations (see DEA models in Emrouznejad (1995-2001)).

On the other hand, SAS is recognized as a one of the lead packages for statistical analysis and as a powerful tool for data base systems in many organizations, both in public and private sectors. A SAS/DEA macro was introduced by Emrouznejad (2000) but the macro is not able to handle the productivity measure such as Malmquist index.

This paper aims to introduce a new application in SAS System for measuring efficiency and productivity of Decision Making Units using DEA and non-parametric Malmquist index techniques.

ABOUT EFFICIENCY AND PRODUCTIVITY

Before starting with the description of SAS code, let us briefly introduce the basic DEA model. The DEA is a linear programming based method for evaluating the relative efficiency of a set of Decision Making Units (DMUs). Unit efficiency can be defined as the capacity of yielding a given level of outputs (products or services) minimizing the quantity of inputs (resources). The DEA transforms multiple input and output measures into a single estimate of efficiency (Boussofiane, Dyson and Thanassoulis, (1991)). Assume a set of observed DMUs, {DMU j; j=1,...,n}, associated with m inputs, {x_{ij} ; i=1,...,m}, and s outputs, {y_{rj} ; r=1,...s}. In the method originally proposed by Charnes, Cooper, Rhodes (1978) relative efficiency of the DMU j_0 can be measured as in model (1).

Model 1. Input oriented - DEA measure

$$
\begin{aligned}
&\text{Min } h\\
&\text{s.t.}\\
&\quad \sum_j \lambda_j x_{ij} + S_i^+ = h x_{ij_0} && \forall i\\
&\quad \sum_j \lambda_j y_{rj} - S_r^- = y_{rj_0} && \forall r\\
&\quad S_i^+ ,\ S_r^- \geq 0 && \forall i, \forall r\\
&\quad \lambda_j \geq 0 && \forall j.
\end{aligned}
$$

Where;
x_{ij} = the amount of the i^{th} input at DMU j,
y_{rj} = the amount of the r^{th} output from DMU j and
j_0 = the DMU to be assessed.

Further, Malmquist index (Färe, et al (1992)) is a productivity measure than can be decomposed to technical change and efficiency change.

With using Malmquist input-oriented productivity index, it is possible to decompose this total productivity change between the two periods into technical change and efficiency change. The input based Malmquist productivity index could be formulated as:

$$M_i^{t+1}(x^t, y^t, x^{t+1}, y^{t+1}) = \left[\frac{D_i^t(x^{t+1}, y^{t+1})}{D_i^t(x^t, y^t)} \times \frac{D_i^{t+1}(x^{t+1}, y^{t+1})}{D_i^{t+1}(x^t, y^t)} \right]^{1/2}$$

Where D_i is the input distance function and $M_i^{t+1}(x^t, y^t, x^{t+1}, y^{t+1})$ is the productivity of the most recent production unit, i.e. A(t+1), using period t+1 technology relative to the earlier production unit, i.e. A(t), with respect to t technology. A value greater than unity will indicate positive total factor productivity growth between the two periods. Following Färe *et al.* (1995) an equivalent way of writing this index is:

$$M = \Delta TECH \times \Delta EFF$$

Where;

$$\Delta EFF = \frac{D_i^{t+1}(x^{t+1}, y^{t+1})}{D_i^t(x^t, y^t)}$$

$$\Delta TECH = \left[\frac{D_i^t(x^{t+1}, y^{t+1})}{D_i^{t+1}(x^{t+1}, y^{t+1})} \times \frac{D_i^t(x^t, y^1)}{D_i^{t+1}(x^t, y^t)} \right]^{1/2}.$$

In this view M, the Malmquist total factor productivity index, is the product of a measure of technical progress, ΔTECH, as measured by shifts in a frontier at period t+1 and period t (average geometrically) and a change in efficiency, ΔEFF, over the same period.

In order to calculate these indexes it is necessary to solve several sets of linear programming problems. Assume there are n DMUs and that each DMU consumes varying amounts of m different inputs to produce s outputs in each period t. The j^{th} DMU, in period t, is therefore represented by the vectors (x_j^t, y_j^t). For example Model 2 measure the distance function of observed

DMU j_0 at period t from the frontier built on the observation at Period t+1.

Model 2. Linear programming models for calculation of the Malmquist index.

$$[D_i^t(x_{t+1}, y_{t+1})]^{-1} = Min\ \phi$$
s.t.
$$\sum_j \lambda_j x_{ij}^t \le \phi x_{ij0}^{t+1} \qquad \forall i$$
s.t.
$$\sum_j \lambda_j y_{rj}^t \ge y_{rj0}^{t+1} \qquad \forall r$$
$$\lambda_j \ge 0 \qquad \forall j$$

SAS / MALM

The SAS/MALM introduced in this paper provides a powerful management tool for assessing both efficiency and productivity of organizations in SAS system. The program can handle both input minimization and output maximization. Further, it can calculate the input and output Malmquist index and its components. To enhance model there are several parameters. Users familiar with SAS can add their own features to build other DEA models. Users not familiar with SAS need only to run the program with their model specification prior running the system.

The SAS/MALM requires two initial data sets that contain name of variable and data file for observed units. The data describing inputs/outputs must be presented in the format that variables appear in columns and units in rows and saved as SAS data set. The program has the ability to accommodate unlimited number of inputs/outputs with unlimited number of DMUs. The only limitation is the memory of computer used to run the SAS/MALM.

The SAS/MALM software then converts data sets to a selected DEA model. Based on the data and parameters specified in the SAS/MALM, the code first create the usual linear program then it use "Proc LP" to solve the model. Results then will be transferred to report files.

The SAS/MALM produces a table of efficiencies of DMUs. It also supplies much other valuable information including lambda and slack values in primal and weights in dual DEA models. These information are very useful for analyzing the inefficient units, where the source of inefficiency comes from and how could improve an inefficient unit to the desired level.

In the rest of this paper the procedure of implementation of SAS/MALM are explained.

The SAS/MALM runs three sections for data handling (**`%data1`** and **`%data2`**), model building (**`%duality`** and **`%dea`**) and report writing (**`%report`**).

DATA HANDLING (%DATA1 AND %DATA2)

This part of SAS/MALM reformats the data to a suitable form that can be used in SAS/OR. The SAS/MALM requires two data sets including name of variable and data file. In data file, variables must be presented in columns and units in rows. The unit names must start with a letter and may contain up to 50 characters and must be listed in the first column of the data file. Period should be recorded in the second column of the data file. The other columns are including numeric values of variables. These variables can be entered into the file in any order.

The variable name file should have two columns. The first column contains the name of variables and the second column contains type of variable (input or output). An example of variable name file is:

File: varname.sd2

```
VARIABLE   IOTYPE
Var1        INPUT
Var2        INPUT
Var3        OUTPUT
Var4        OUTPUT
```

There are six parameters prior to calling the data macro:
_LibName: name of directory where all files were saved,
_DataF: name of data file,
_VarF: name of variable file,
_Orienta: to select the orientation of the model, it can be "OUTPUTMAX" or "INPUTMIN",
_period1: first period for Malmquist index and
_Period2: second period for Malmquist index.
SAS procedure for data handling is as follows.

SAS procedure for data handling

```
* Macro for transferring data files to suitable
format for DEA models;
%Macro data1;
 data _TypeF;
 format _type_ $3. ;
 format IOTYPE $7. ;
 _type_='LE' ; IOType='OUTPUT'; output;
 _type_='LE' ; IOType='INPUT' ; output;
 IOType='OBJ'; output;
 run;
 data &_VarF;
 set &_varF;
 Variable=upcase(variable);
 run;
 proc sort
data=&_VarF(where=(UPCASE(IOType)='INPUT'))
out=_VarFi;
 by Variable;
 run;
 data _null_;
 set _VarFi;
 call sympt('_nInput',_n_);
 run;
 data _VarFi;
 do i= 1 to &_nInput;
  Link Read;
  _VarName=compress('X'||i); output;
 end;
 Read: set _VarFi;return;
 drop i;
 run;
 proc sort
data=&_VarF(where=(UPCASE(IOType)='OUTPUT'))
 out=_VarFo;
 by Variable;
 run;
 data _null_;
 set _VarFo;
 call sympt('_nOutput',_n_);
 run;
 data _VarFo;
 do r= 1 to &_nOutput;
 Link Read;
 _VarName=compress('Y'||r); output;
 end;
 Read: set _VarFo;return;
 drop r;
 run;
 proc sort data=&_DataF
 out=&_DataF;
 by DMU;
 run;
 proc summary data=&_DataF;
 output out=_Dmu1F(keep=DMU);
 by DMU;
 run;
 data _null_;
 set _Dmu1F;
 call sympt('_nDmu',_n_);
 run;
 data _DmuF;
 do j= 1 to &_nDmu;
  Link Read;
  _VarName=compress('DMU'||j); output;
 end;
 Read: set _Dmu1F;return;
 drop j;
```

```
 run;
 data _DataF1;
 merge &_DataF _DmuF ;
 by DMU;
 run;
 data _VarF1;
 set _VarFi _VarFo;
 run;
 proc sort data=_VarF1;
 by variable;
 run;
 data _ObjF; _VarName='TETA';
 Variable='Objective';
 IOType='OBJ';
 if &_Orienta='INPUTMIN'
  then do;
  _type_='MIN';
  end;
 if &_Orienta='OUTPUTMAX'
  then do;
  _type_='MAX';
  end;
 output;
 RUN;

 data _D&_Per1.;
 set _DataF1(where=(Period=&_Per1));
 run;
 data _D&_Per2.;
 set _DataF1(where=(Period=&_Per2));
 run;
 data _TypeF;
 set _TypeF;
 IOTYPE=UPCASE(IOTYPE);
 run;
 %do t=&_Per1 %to &_Per2;
 proc transpose data=_D&t.(drop=DMU)
 out=_D&t.t(rename=_Name_=Variable);
 id _VarName;run;
  proc sort data=_D&t.t;
 by variable;
 run;
 data _VarF2;
 merge _VarF1(in=IO) _D&t.t (in=Dt) ;
 by Variable; if Dt and IO;
 run;
  proc sort data=_VarF2;
 by IOType;
 run;
 proc sort data=_TypeF;
 by IOType;
 run;
 data _F4&t.;
  merge _VarF2(in=Dt) _TypeF;
  by IOType;
  Period=&t.;
  if Dt;
  run;
 data _F4&t.;
  set _F4&t. ;
  if UpCase(IOTYPE)='OUTPUT' then
  do;
   %do j0=1 %to &_nDmu ;
   DMU&j0=-DMU&j0;
   %end;
  end;
  run;
 %end;
 data _Model01;
 set _F4&_Per1.;
 _Kind='T1xt1yt1';
 run;
 data _Model02;
 set _F4&_Per2.;
 _Kind='T2xt2yt2';
 run;
%mend data1;
* Macro for transfering data files to suitable
format for Malmquist index;
%macro data2(Uj0);
 data _Model0x(keep=_varname &Uj0
rename=(&Uj0=TempUj0));
 set _F4&_Per2.;
 run;
 data _Model0y; merge _Model0x _F4&_Per1.;
 by _VarName;
 run;
 data _Model03(drop=TEMPUj0);
 set _Model0y;
 if UPCASE(IOTYPE)='INPUT' then &Uj0=TEMPUj0;
 _Kind='T1xt2yt1';
 run;
 data _Model0x(keep=_varname &Uj0
rename=(&Uj0=TempUj0));
 set _F4&_Per2.;
 run;
 data _Model0y;
 merge _Model0x _F4&_Per1.;
 by _VarName;
 run;
 data _Model04(drop=TEMPUj0);
 set _Model0y;
 if UPCASE(IOTYPE)='OUTPUT' then &Uj0=TEMPUj0;
 _Kind='T1xt1yt2';
 run;
 data _Model0x(keep=_varname &Uj0
rename=(&Uj0=TempUj0));
 set _F4&_Per1.;
 run;
 data _Model0y;
 merge _Model0x _F4&_Per2.;
 by _VarName;
 run;
 data _Model05(drop=TEMPUj0);
 set _Model0y;
 if UPCASE(IOTYPE)='INPUT' then &Uj0=TEMPUj0;
 _Kind='T2xt1yt2';
 run;
 data _Model0x(keep=_varname &Uj0
rename=(&Uj0=TempUj0));
 set _F4&_Per1.;
 run;
 data _Model0y; merge _Model0x _F4&_Per2.;
 by _VarName;
 run;
 data _Model06(drop=TEMPUj0);
 set _Model0y;
 if UPCASE(IOTYPE)='OUTPUT' then &Uj0=TEMPUj0;
 _Kind='T2xt2yt1';
 run;
 data _Model0x(keep=_varname &Uj0
rename=(&Uj0=TempUj0));
 set _F4&_Per1.;
 run;
 data _Model0y;
 merge _Model0x _F4&_Per2.;
 by _VarName;
 run;
 data _Model07(drop=TEMPUj0);
 set _Model0y;
 &Uj0=TEMPUj0;
 _Kind='T2xt1yt1';
 run;
 data _Model0x(keep=_varname &Uj0
rename=(&Uj0=TempUj0));
 set _F4&_Per2.;
 run;
 data _Model0y;
 merge _Model0x _F4&_Per1.;
 by _VarName;
 run;
 data _Model08(drop=TEMPUj0);
 set _Model0y;
 &Uj0=TEMPUj0;
 _Kind='T1xt2yt2';
 run;
%mend data2;
```

MODEL BUILDING

This part of SAS/MALM builds the requested DEA model in the format suitable for SAS/OR and for purpose of calling "Proc LP". There is one parameter prior calling the procedures:

```
_Orienta
```

For example, for solving a standard input minimization model the user should set the parameters to:

```
_Orienta='INPUTMIN';
```

and for solving output maximization model the user should set the parameters to:

```
_Orienta='OUTPUTMAX';
```

SAS procedure for model building is as follows.

SAS procedure for model building

```
* Macro for preparation dual model;
%macro Duality(_PrimF, _DualF, _Ind);
 proc sort data=&_PrimF out=Temp1;
 by _row_;
 run;
 proc transpose data =Temp1 out=Temp2;
 id _row_ ;
 run;
 data &_DualF(rename=(Teta=_RHS_ _name_=_row_));
 set Temp2;
 format _type_ $3.;
 if _name_='_RHS_' then _name_='TETA';
 if Teta=. then Teta=0;else Teta=-Teta;
 if substr(_name_,1,2)='FI'
  then _type_='EQ';
  else _type_='GE';
 if upcase(_name_)='TETA' then _type_='MIN';
 run;
%mend Duality;

* Macro for prepartion DEA models;
%macro DEA(DMUj0, M);
 data _ModelP1(rename=(_VarName=_row_)
drop=IOTYPE PERIOD variable);
  set _Model0&M;
  if &_orienta='INPUTMIN'
  then do;
   if UPCASE(IOType)='INPUT'
   then do;
    FI=-&DMUj0;
    _RHS_=0 ;
   end;
   if UPCASE(IOType)='OUTPUT'
   then do;
    FI=0;
    _RHS_=&DMUj0 ;
   end;
  end;
  if &_orienta='OUTPUTMAX'
  then do;
   if UPCASE(IOType)='OUTPUT'
   then do;
    FI=-&DMUj0;
    _RHS_=0 ;
   end;
   if UPCASE(IOType)='INPUT'
   then do;
    FI=0;
   _RHS_=&DMUj0 ;
   end;
  end;
  run;
 %Duality(_ModelP1, _ModelW1,'LE');
 data _ModelR1;
 set _Modelw1;
 if _row_='FI'
 then do;
  if &_Orienta='INPUTMIN' then _rhs_=-1;
  if &_Orienta='OUTPUTMAX' then _rhs_=1;
  end;
 run;
 proc LP data= _Modelr1
 primalout=RP dualout=RD MAXIT2=100000;
 run;
 data RP;
 set RP;
 _DMUj0="&DMUj0";
 _KIND=&M.;
 run;
 data RD;
 set RD;
 _DMUj0="&DMUj0";
 _KIND=&M.;
 run;
 proc datasets nolist force;
 append base=sasmalm.MRP data=RP;
 run;
 proc datasets nolist force;
 append base=sasmalm.MRD
 data=RD;
 run;
%mend dea;
```

SAS/MALM EFFICIENCY REPORT WRITING (%REPORT)

The SAS/MALM results are including table of efficiencies of DMUs in two formats sorted by the most efficient DMU and sorted alphabetically by the name of DMUs. All other information like slacks and lambdas are saved in file called "Report1". All information on the dual to each model including weights are saved in file called "Report2".

SAS procedure for report writing is as follows.

SAS procedure for report writing

```
* Macro for reporting the results;
%macro report;
 data _KindF;
 _kind=1; _tKind='T1xt1yt1'; output;
 _kind=2; _tKind='T2xt2yt2'; output;
 _kind=3; _tKind='T1xt2yt1'; output;
 _kind=4; _tKind='T1xt1yt2'; output;
 _kind=5; _tKind='T2xt1yt2'; output;
 _kind=6; _tKind='T2xt2yt1'; output;
 _kind=7; _tKind='T2xt1yt1'; output;
 _kind=8; _tKind='T1xt2yt2'; output;
 run;
 proc sort
data=sasmalm.mrp(rename=(_Dmuj0=_VarName));
 by _VarName;
 run;
 proc sort data=_DmuF;
 by _VarName;
 run;
 data report1(keep=DMU _value_ _kind);
 merge _DmuF sasmalm.mrp(where=( _var_='TETA'));
 by _VarName;
 run;
 data report1(drop=_value_);
 set report1;
 if &_Orienta='INPUTMIN' then eff=-_value_;
 if &_Orienta='OUTPUTMAX' then eff=1/(_value_);
 run;
 proc sort data=report1;
 by _Kind;
 run;
 data report1(drop=_kind);
 merge report1 _KindF;
 by _kind;
 run;
 proc sort data=report1;
 by DMU;
 run;
 proc transpose data=report1
out=sasmalm.report(drop=_name_);
 by DMU;
 id _tkind;
 run;
%mend report;
```

SAS/MALM MACRO

To make the system as ease as possible the “**%sasmalm macro**” put all the above code together.

```
%macro sasmalm;
 libname sasmalm &_libname;
 %data1;
 proc datasets nolist library=sasmalm ;
 delete MRP MRD;
 run;
 %do j_Note=&_nDmu %to 1 %by -1;
  %do M=1 %to 8;
  %let Uj0=DMU&j_Note;
  %data2(&Uj0);
  %DEA(&Uj0, &M);
  %end;
 %end;
 %report;
%mend sasmalm;
```

```
* Parameter definitions and running SAS/MALM;
%let _libname='c:\sasmalm'; * Name of directory;
%let _DataF=sasmalm.Data2; * Name of data file;
%let _VarF=sasmalm.Var2;  * Name of variable
file;
%let _Orienta='INPUTMIN'; * Alternative
selection is 'OUTPUTMAX';
%let _Per1=1;   * First period ;
%let _Per2=2;   * second period;
       % sasmalm;
```

In the above code the **"%sasmalm macro"** is used to manage all previously explained codes including data handling, model building and report writing. To get the result, user needs to set up the parameters and run only one statement:

%SASMALM;

CONCLUSION

Today many organizations recognized SAS as a one of the lead packages for data base system and statistical analysis. In particular, optimization procedures in SAS/OR are exposed to the user in a variety of places such as "Proc LP" and "Proc NLP". Therefore many applications such as neural network, control project management (Cohen and Meanor (1995)) and SAS for DEA user (Emrouznejad(2000)) have been developed. SAS/MALM as introduced in this paper is a new application in SAS/OR that is a powerful managerial tool for measuring the efficiency and productivity of Decision Making Units.

The SAS/MALM application implemented in this paper has no limitation on the input and output variables or number of DMUs. The only limitation is the memory and disk space of the computer uses. It is flexible to add other DEA model.

A SAS/MALM for end user is as ease as to run one statement:

%sasmalm.

The SAS/MALM report's files can directly feed to other SAS routines for further analysis:

REFERENCES

Boussofiane, A, R. Dyson and E Thanassoulis, (1991), "Applied data envelopment analysis", *European Journal of Operational Research* vol. 52, pp. 1-15.

Charnes, A., W. W. Cooper, and E. Rhodes. (1978), "Measuring the Efficiency of Decision Making Units", *European Journal of Operations Research* 2(6) pp 429-44.

Cohen, Marc-David and Phil Meanor (1995), *QSIM Application Discrete Event Queuing Simulation, Release 6.11*, Cary, NC: SAS Institute Inc.

Cooper, W. W., M. L. Seiford and K. Tone (1999), *Data Envelopment Analysis: A comprehensive text with models, applications, references and DEA solver software*, Kluwer Academic Publishers, Boston.

Emrouznejad, A (2000) "An Extension to SAS/OR for Decision Support System," *SAS Users Group International, 25th Annual Conference*, April 9-12, 2000, Indiana Convention Centre, Indianapolis, Indiana, USA.

Emrouznejad, A. (1995-2001) "Ali Emrouznejad's Data Envelopment Analysis Homepage", *http://www.deazone.com/*, Warwick Business School, Warwick University, UK

Färe, R. Sh. Grosskopf and P. Roos (1995), "Productivity and quality changes in Swedish pharmacies," *International Journal of Production Economics,* 39(1,2): 137-147

Färe, R., Sh. Grosskopf, B. Lindgren and P. Roos (1992), "Productivity changes in Swedish pharmacies 1980-1989: A non-parametric Malmquist approach," *Journal of Productivity Analysis*, 3:85-101.

Farrell, M. J. (1957), "The Measurement of productive efficiency," *Journal of Royal Statistical Society,* Series A (120): 253-290.

Funkuyama, H. (1995), "Measuring efficiency and productivity growth in Japanese banking: a non-parametric frontier approach," *Appl. Financial Econ*., 5: 95-107.

SAS Institute Inc. (1989). *SAS/OR User's Guide, Version 6*, First Edition, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1993). *SAS/OR User's Guide, Project Management, Version 6*, First Edition, Cary, NC: SAS Institute Inc.

Trevor Kearney (1999), "Advances in Mathematical Programming and Optimization in the SAS System", *SUGI24 Proceedings*, Cary, NC: SAS Institute Inc.

CONTACT INFORMATION

Author:

Ali Emrouznejad
http://www.deazone.com
Email: a_emrouznejad@hotmail.com