Simulation of Propensity Scoring Methods

Dee H. Wu, Ph.D, David M. Thompson, Ph.D., David Bard, Ph.D.
University of Oklahoma Health Sciences Center, Oklahoma City, OK

ABSTRACT

In certain clinical trials or observational studies, proper random assignment of treatment and control groups is not always possible, so that selection bias may become an issue. Recent efforts to address issues of nonrandom assignment, including a class of methods known as 'Propensity Scoring,' are alternatives to reduce bias in the estimation of treatment effects when assignment is not random. The original technique was described by Rosenbaum and Rubin in 1983. However, we have found that much of current literature describes the technique using a large number of mathematical constructs. The multiplicity of mathematical approaches reduces the techniques' accessibility to users from different interdisciplinary departments. Failure to understand the statistical method deters proper usage. We will first describe the technique through diagram and simulation to improve teaching and understanding the material intuitively in SAS. We hope that the introduction permits readers to enhance their own understanding of the relevant mathematical formulism presented in the literature.

INTRODUCTION

On returning from a national SAS users group meeting, we presented the Propensity Scoring Method to our local student/faculty SAS group. It was apparent from the group's response that those who did not attend the national meeting were only able to follow parts of the complex presentation. We decided to create a simulation and visual presentation of the method in SAS for teaching and to put the problem in the hands of students.

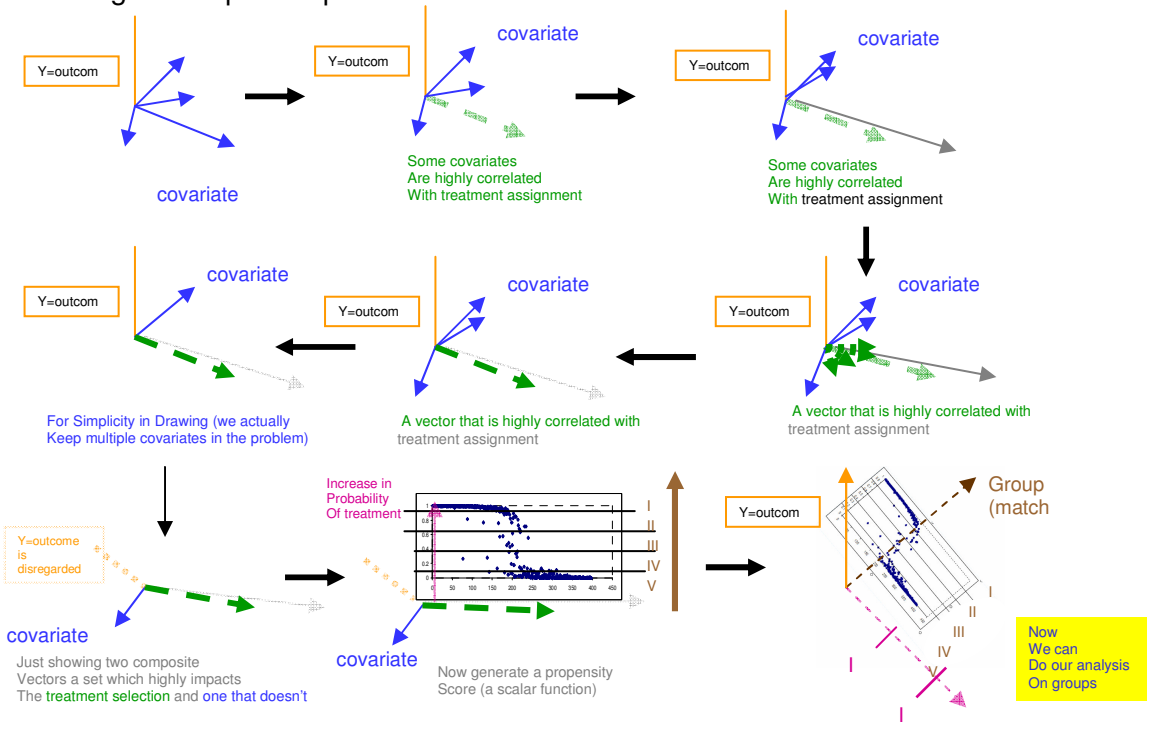

Figure 1: Overview of the propensity scoring method in pictures in a nutshell.

Our first goal was to describe the mathematics purely through images (see figure 1). The methods' multiple steps make propensity scoring methods (PSM) difficult to grasp. PSM involves projection of vectors, the assumption that a vector of covariates is highly correlated with treatment assignment, the use of logistic regression, some algorithm for matching or stratification (or classification such as design tree), and the construction of general linear models. Only when we created these pictures did intuition about the method became more apparent to us.

We generated a test set based on 4 example models that we designed and selected to show the technique's strengths and weaknesses. The simulation procedure is described below

Methods:

1. Generate four simulation sets (with covariates X and output Y)

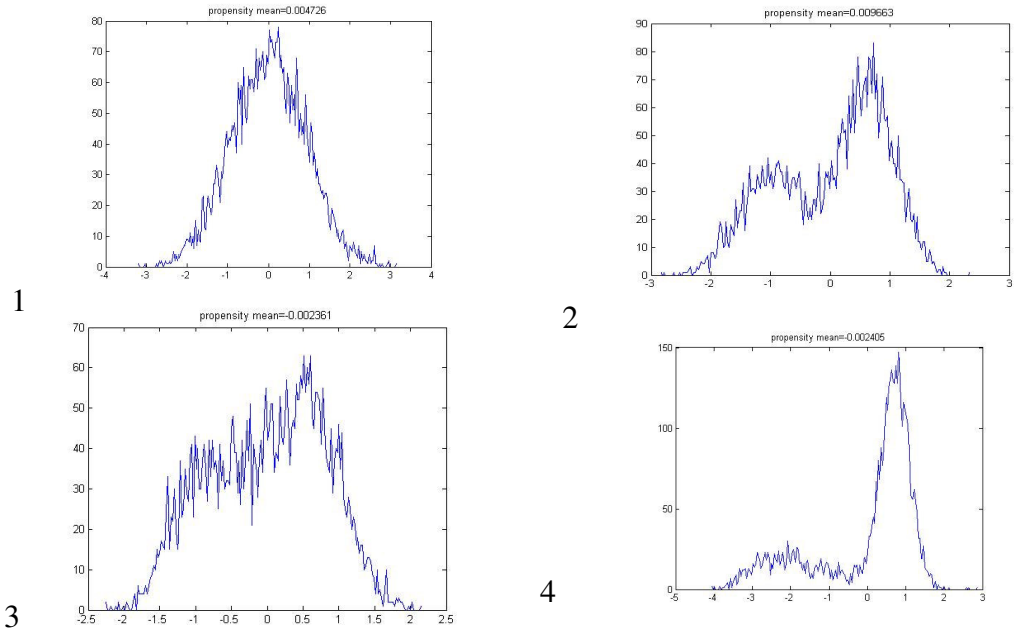

Figure 2: Histograms of four exemplary distributions. Although we produced these distributions with N=5000 for display, we simulated test cases with N=500 to create larger sensitivity in the estimates.

The parameters for the four cases were simulated in Matlab but can be easily run in SAS:

case 1: x1 = GenerateNHumpRand(x,[-4 4],[0 1 0],[0 0 0], [1 1 1]);
case 2: x1 = GenerateNHumpRand(x,[-4 4],[0.4 0 0.6],[-1 0 1], [0.5 0 0.3]);
case 3: x1 = GenerateNHumpRand(x,[-4 4],[0.3 0.2 0.5],[-1 0 1], [0.2 0.3 0.3]);
case 4: x1 = GenerateNHumpRand(x,[-4 4],[0.3 0 0.7],[-3.5 0 1], [2 .1 0.1]);

The inputs to the function GenerateNHumpRand are:

- The first vector [-4 4] is the permissible range from which the simulation draws the x values from a standard normal distribution.
- The second vector's size indicates the distribution's number of humps, and the fractional multipliers specify the size of the observation
- The third vector specifies locations for the means of the distribution,
- The fourth vector specifies standard deviations.

2. Create 3 covariates (x2, x3, x4) that are uncorrelated with x1, which was generated above. Although the vector x1 may in reality be a combination of other vectors, we reduced x1 to a single one for demonstration simplicity.

Pearson Correlation Coefficients, N = 500
Prob > |r| under H0: Rho=0

	y	x1	x2	x3	x4	propensity
y	1.00000	0.61654 <.0001	0.27374 <.0001	0.47571 <.0001	0.59396 <.0001	0.66762 <.0001
x1	0.61654 <.0001	1.00000	0.03578 0.4247	0.35222 <.0001	0.02003 0.6550	0.92131 <.0001
x2	0.27374 <.0001	0.03578 0.4247	1.00000	0.00116 0.9794	0.02229 0.6190	0.22981 <.0001
x3	0.47571 <.0001	0.35222 <.0001	0.00116 0.9794	1.00000	-0.00268 0.9523	0.38666 <.0001
x4	0.59396 <.0001	0.02003 0.6550	0.02229 0.6190	-0.00268 0.9523	1.00000	0.05991 0.1810
propensity	0.66762 <.0001	0.92131 <.0001	0.22981 <.0001	0.38666 <.0001	0.05991 0.1810	1.00000

3. Calculate a propensity score based a linear combination (weights sum to 1) of the covariates X1 to X4, along with a small random normal error term The variable 'propensity score was generated by a linear combination of the covariates but could have been generated by any function.

*propensity = 0.79*x1+0.15*x2+0.05*x3+0.01 *x4+0.3*Normal(0,1);*

4. Assign to Treatment group based on the propensity score. Note we use run a standard random normal probability with mean 0 and std=1 to assign treatment group (i.e. x1 has a large influence on setting the treatment group). Note, we used the inverse Probit transformation to control the treatment/control ratio (proportions) and for generating the proper threshold for classifying the treatment covariate. Notice z1 has a large influence on the treatment classification. The scores we set up typically had ranges of -2 to 2.

5. Create observation vector Y for each of the test cases

$$Y= a1*x1 + a2*x2 + a3*x3 + a4*x4 + b*T + c1*T*x1 + c2*T*x2 + c3*T*x3 + c4*T*x4 + 0.4*Norm(0,1)$$

where, a1=0.28,a2=0.13,a3=0.23,a4=0.352; b=0.4; c1=c2=c3=0.2,c4=0.02

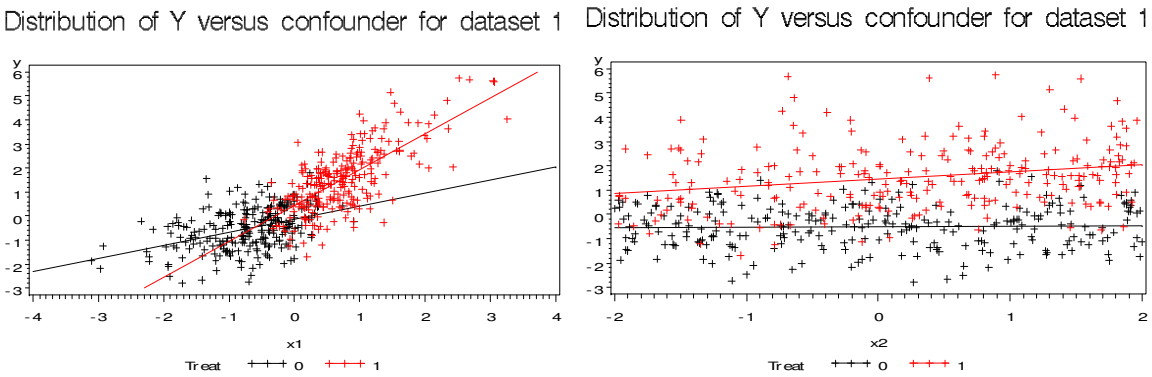

Figure 3: The treatment set and control group are plotted against x1. In many applications, several covariates may be associated with treatment assignment, not just a single one like x1. x1 can represent a 'modeled' linear combination of covariates.

Note that, Figure 3 illustrates heterogeneity of regression that would challenge conventional ANCOVA methods but may not affect PSM.

Results and Discussion:

Distribution	Description	Niave	Quintile Method	Full Model	Error Quint	Error Full	Delta Quint	Delta Full
1	1lump	1.62	0.399	0.375	-0.3%	**-6.3%**	0.001	-0.025
2	2lump	1.45	0.411	0.471	2.7%	**17.8%**	0.011	0.071
3	3lumpbroad	1.13	0.324	0.378	**-19.0%**	-5.5%	-0.076	-0.022
4	2assymetric	1.64	0.67	0.28	**67.5%**	-30.0%	0.27	-0.12

Table 1: We compare the Propensity Quintile method vs the Full Model

Note that in cases where the propensity scores behave "well" and cover a wide range of values, the propensity quintile method outperformed the Full regression model:

```
model y = treat x1 x2 x3 x4 treat*x1 treat*x2 treat*x3 treat*x4
```

However, when the propensity scores are more asymmetrically distributed or mixed, the propensity method was less useful. As we taught ourselves, the efficacy of the propensity scoring technique depended on its ability to create a well-behaved function, and on the 'matching' technique. Matching can be improved with better classification schemes; we performed a simple quintile stratification.

Conclusion:

Our approach to demonstrating the Propensity Scoring technique by pictorial description over mathematical formulation was better received by faculty and students, helped our understanding,

and built collaboration. This demonstration provided a tool that students and faculty can use to better understand the propensity technique. This work can be extended to the multiple varieties of discriminant/classification schemes available for matching. Regardless, users of the technique should beware of its pitfalls and challenges once they have understood the base method.

The Propensity Method in Parsimonious Mathematical Formulism.

Now we return to the mathematics, as extracted from our understanding and from an online search for the topic.

$$Y_i = \alpha + \tau W_i + X_i^{'}\beta + \varepsilon_i$$

- α , the grand mean, is the average treatment effect (in condition that assignment to the treatment or control group is random)
- τ is the treatment effect
- $W_i = 1$ for individuals in the treated group. $W_i = 0$ for those in the control group.
- Y_i is the observed outcome of individual i
- X_i includes a set of observed characteristics (covariates), some of which affection selection into treatment.
- ε_i is the error term denoting unobserved characteristics.
- Z_i is an observed variable that affects selection into the treatment.

Matching is necessary because of dependence between ε_i and W_i that is due to a set of observed covariates, X_i, that are associated with selection into the treatment. Matching specifically sets:

$$E(\varepsilon_i \mid W_i, X_i, Z_i) = E(\varepsilon_{it} \mid X_i, Z_i).$$

We want to estimate $W_i \mid Z_i$ (the probability of Wi given Zi) by constructing an independent vector that is orthogonal to it and is therefore free from selection or other bias. The matching condition can be described as: $(Y_{i1}, Y_{i0}) \perp W_i \mid Z_i$ and provides a vector from which we can calculate unbiased estimates of treatment effects.

Appendix:

(Quintile y = psquintile treat) 213
15:11 Wednesday, August 1, 2007

The GLM Procedure

Dependent Variable: y

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	396.5209575	79.3041915	113.83	<.0001
Error	494	344.1678631	0.6966961		
Corrected Total	499	740.6888206			

R-Square	Coeff Var	Root MSE	y Mean
0.535341	232.8230	0.834683	0.358505

Source	DF	Type I SS	Mean Square	F Value	Pr > F
psquintile	4	389.9494274	97.4873568	139.93	<.0001
Treat	1	6.5715301	6.5715301	9.43	0.0022

Source	DF	Type III SS	Mean Square	F Value	Pr > F
psquintile	4	121.4500828	30.3625207	43.58	<.0001
Treat	1	6.5715301	6.5715301	9.43	0.0022

Parameter		Estimate		Standard Error	t Value	Pr > |t|
Intercept		1.834332340	B	0.08346832	21.98	<.0001
psquintile	1	-2.124842808	B	0.17569486	-12.09	<.0001
psquintile	2	-1.739396969	B	0.16905856	-10.29	<.0001
psquintile	3	-1.501344298	B	0.13235300	-11.34	<.0001
psquintile	4	-1.014376395	B	0.11890682	-8.53	<.0001
psquintile	5	0.000000000	B	.	.	.
Treat	0	-0.399669592	B	0.13013363	-3.07	0.0022
Treat	1	0.000000000	B	.	.	.

(Quintile y = psquintile treat) 213

Table 1: The results from the 1 lump model (distribution 1), note the high convergence with the desired

Full ANOVA model y = treat x1 x2 x3 x4 treat*x1 treat*x2 treat*x3 treat*x4 /solution; 511
15:11 Wednesday, August 1, 2007

The GLM Procedure

Dependent Variable: y

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	9	666.0799359	74.0088818	486.06	<.0001
Error	490	74.6088847	0.1522630		
Corrected Total	499	740.6888206			

R-Square	Coeff Var	Root MSE	y Mean
0.899271	108.8432	0.390209	0.358505

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Treat	1	275.0708747	275.0708747	1806.55	<.0001
x1	1	101.9614193	101.9614193	669.64	<.0001
x2	1	34.3399471	34.3399471	225.53	<.0001
x3	1	6.9045552	6.9045552	45.35	<.0001
x4	1	237.9048111	237.9048111	1562.46	<.0001
x1*Treat	1	1.9901006	1.9901006	13.07	0.0003
x2*Treat	1	7.5437225	7.5437225	49.54	<.0001
x3*Treat	1	0.0113856	0.0113856	0.07	0.7846
x4*Treat	1	0.3531199	0.3531199	2.32	0.1284

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Treat	1	2.2122481	2.2122481	14.53	0.0002
x1	1	0.9602612	0.9602612	6.31	0.0123
x2	1	37.0700548	37.0700548	243.46	<.0001
x3	1	0.0221250	0.0221250	0.15	0.7032
x4	1	238.4653824	238.4653824	1566.14	<.0001
x1*Treat	1	0.0471813	0.0471813	0.31	0.5780
x2*Treat	1	7.5938502	7.5938502	49.87	<.0001
x3*Treat	1	0.0109239	0.0109239	0.07	0.7889
x4*Treat	1	0.3531199	0.3531199	2.32	0.1284

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	0.327794072 B	0.07515835	4.36	<.0001
Treat 0	-0.374758202 B	0.09831764	-3.81	0.0002

Table 2: The results from the 1 lump model (distribution 1) with the full model (including convariates and treatment effects

```
% 08-02-07 Dee Wu, David Thompson
% University of Oklahoma Health Sciences Center
%macro r;
%do i=2 %to 2;
/*create working SAS data step from alump tab-delimited text file*/
%let indata="f:\wu\latest\propensity&i.lump.txt";
data in;
 infile &indata firstobs=2 dlm='09'x;
 input num propensity  x1     x2    x3    x4
a1    a2    a3    a4
b     c1    c2    c3    c4 d1 sort1 sort2 sort3 Treat y T2 y2 Y3;
run;

goptions reset=all;
title "Distribution of Y versus confounder x1 for dataset &i";
proc gplot data=in;
  symbol v=plus i=rl;
  plot y*x1=Treat;
run;

goptions reset=all;
title "Distribution of Y versus confounder x2 for dataset &i";
proc gplot data=in;
  symbol v=plus i=rl;
  plot y*x2=Treat;
run;
goptions reset=all;

proc corr data=in;
var y x1 x2 x3 x4 propensity;
run;

proc univariate data=in;
title "Distribution of Y versus confounder x1 for dataset &i";
var y;
histogram y;
run;

/*This step calculates the mean and SD of the outcome Y
for the two observed treatment groups.  While our initial attempts
at discerning true group membership in a mixture model used this
information, it's less important here.*/
proc means data=in mean std;
  class treat;
  var y;
  ods output summary=treatmeans;
run;

/*Logistic regression obtains PROPENSITY SCORES
in the form of predicted probabilities that an observation
is assigned to treatment group 1, conditional on its observed
covariates.  The regression function considers just the covariates (including
potential confounders), not the outcome Y.*/
proc logistic data=in;
  model treat (event='1')= x1 x2 x3 x4;
  output out=predprobs p=phat;
run;
```

```
/*Observation can be grouped on the basis of the predicted probabilities
that are output from the previous step.

One approach simply dichotomizes the grouping. Because phat=p(treat=1),
predgroup is 1 if phat is high (.5 or higher), and 2 if phat is less
than .5*/
data predict1;
  set predprobs;
  predgroup=2-(phat ge .5);
run;
/*A more flexible approach groups observations on the basis of quantiles.*/
proc rank data=predprobs groups=5 out=rank;
      ranks psquintile;
      var phat;
run;

proc freq data=rank;
      tables psquintile*treat;
run;

proc sort data=rank;
  by descending treat;
run;
/*Checking associations among covariates and quantile assignments.*/
proc glm data=rank order=data;
title('model x1 x2 x3 x4 = psquintile treat psquintile*treat');
      class psquintile treat;
      model x1 x2 x3 x4 = psquintile treat psquintile*treat;
run;

/*THIS INTERACTION MODEL PRODUCES DATA S1 WITH BETA ESTIMATE*/
proc glm data=rank order=data;
title('model y = psquintile treat psquintile*treat');
      class psquintile treat;
      model y = psquintile treat psquintile*treat / solution;
    ods output parameterestimates=s1 (where=(substr(parameter,1,5)='Treat'));
run;

/*THIS NO INTERACTION MODEL THAT ADJUSTS FOR THE PROPENSITY SCORE'S
QUINTILE PRODUCES DATA S2 WITH BETA ESTIMATE*/
proc glm data=rank order=data;
title('model y = psquintile treat');
      class psquintile treat;
      model y = psquintile treat / solution;
    ods output parameterestimates=s2 (where=(substr(parameter,1,5)='Treat'));
run;

/*THIS NAIVE MODEL MAKES NO ADJUSTMENT AND RECORDS BETA ESTIMATE
IN DATA S3*/
proc glm data=rank order=data;
title('model y=treat');
      class  treat;
      model y =  treat / solution;
    ods output parameterestimates=s3 (where=(substr(parameter,1,5)='Treat'));
run;
```

```
/*WEIGHTING THE NAIVE MODEL BY THE PROPENSITY SCORE*/
proc glm data=rank order=data;
  weight phat;
  class treat;
  model y=treat / solution;
  ods output parameterestimates=s4 (where=(substr(parameter,1,5)='Treat'));
run;

/*TREATING THE PROPENSITY SCORE AS A SURROGATE FOR INFORMATION ON THE
COVARIATES*/
proc glm data=rank order=data;
  class treat ;
  model y=treat phat / solution;
  ods output parameterestimates=s5 (where=(substr(parameter,1,5)='Treat'));
run;

/*A final method uses IPTW estimators by constructing
inverse weights from original predicted probabilities,
then using these in another logistic regression.*/
data weights;
  set predprobs;
  if treat=1 then w=1/phat;
  if treat=0 then w=1/(1-phat);
run;
/*and runs the regression model with IPTW
taking place of covariates that might confound
exposure of interest.*/
proc glm data=weights order=data;
  class treat;
  model y = treat w / solution;
  ods output parameterestimates=s6 (where=(substr(parameter,1,5)='Treat'));
run;

/*INSTEAD OF USING THE IPTW ESTIMATOR AS
A SURROGATE FOR THE COVARIATES, USE IT AS A WEIGHT.*/
proc glm data=weights order=data;
  class treat;
  WEIGHT W;
  model y = treat / solution;
  ods output parameterestimates=s7 (where=(substr(parameter,1,5)='Treat'));
run;

/*THIS STEP COLLECTS THE BETA ESTIMATES IN A SINGLE DATASET and applies a
format
to identify the models that produced the estimates. */
proc format;
  value methf 1='model y = psquintile treat psquintile*treat'
              2='model y = psquintile treat'
                    3='unadjusted model y=treat'
                    4='unadjusted model weighted by propensity score'
                    5='model y=treat phat'
                    6='IPTW estimator as surrogate for covariates model
y=treat w'
                    7='IPTW estimator as weight in model y=treat'
                    ;
data results;
  set s1 s2 s3 s4 s5 s6 S7;
```

```
  if estimate ne 0;
  method+1;
  format method methf.;
  drop dependent biased;
run;

%end;
%mend r;
%r;
quit;
```