Paper

Get Better WOE of Scorecard by Genetic Algorithm

Xia Ke Shan, iFRE Inc., Beijing, China

Peter Eberhardt, Fernwood Consulting Group Inc., Toronto, Canada

Matthew Kastin, NORC at the University of Chicago, Hummelstown, Pennsylvania

ABSTRACT

Scorecard is a very important risk management tool in credit card business. When a client start to apply to a credit card in a bank, if his scorecard was lower than a cutoff score (e.g. 520), then the bank is going to reject his apply, vice versa. The WOE (a.k.a weight of evidence) plays the most important role in how to build a better scorecard to distinct the bad person and good person. This paper is trying to get better WOE by solving an optimization problem via genetic algorithm.

INTRODUCTION

What is the WOE? What does the better WOE look like?

WOE measures the strength of grouped attributes of a variable in separating good and bad accounts. And it represents predictive power of each attribute. Here is an example to demonstrate what WOE is (it is taken from the second reference book).Suppose there are two variables. One flags the bad and good person, it has two values: bad and good. Another is age.

Age	Count	Tot Distr	Goods	Distr Good	Bads	Distr Bad	Bad Rate	WOE
Missing	1,000	2.50%	860	2.38%	140	3.65%	14.00%	−42.719
18–22	4,000	10.00%	3,040	8.41%	960	25.00%	24.00%	−108.980
23–26	6,000	15.00%	4,920	13.61%	1,080	28.13%	18.00%	−72.613
27–29	9,000	22.50%	8,100	22.40%	900	23.44%	10.00%	−4.526
30–35	10,000	25.00%	9,500	26.27%	500	13.02%	5.00%	70.196
35–44	7,000	17.50%	6,800	18.81%	200	5.21%	2.86%	128.388
44+	3,000	7.50%	2,940	8.13%	60	1.56%	2.00%	164.934
Total	**40,000**	**100%**	**36,160**	**100%**	**3,840**	**100%**	**9.60%**	

Here, age is divided into six groups. Column Goods stand for the good number in a group. For example, group 18-22 has 3040 good person. The same logic to column Bads, group 18-22 has 960 bad person. Column DistrGood=the good number in a group / all the good number. For example, for group 18-22, DistrGood= 3040/36160=8.41%. The same logic to column DistrBad. After that WOE=ln(DistrGood/DistrBad). For example, for group 18-22, WOE=ln(8.41%/25%)=-108.9.

What does the better WOE look like?

1) The difference between the WOE of groups should be as larger as it could be. The larger the difference between adjacent groups, the higher the predictive ability of this variable. And score would be most different from each group.
2) Column DistrGood and DistrBad generally should be greater than 5%. If it was too small, then there are not enough records to reflect the underlying rule. difference between adjacent groups, the higher the predictive ability of this variable. And score would be most different from each group.
3) For the continuous variable (e.g. age), WOE should be monotonous increase or decrease, better is linear.

4) For the category variable (e.g. sex), there is no requirement for WOE to be monotonous increase or decrease, but better make IV(Information Value) is as bigger as it could be. IV measures the strength of correlation between the variable and Y (odds for good or bad).

   IV= Σ (DistrGoodi - DistrBadi) * ln(DistrGoodi / DistrBadi)

5) Make as many groups as you can, more groups can keep more details of original variable, and make the grouped variable has higher IV.

EXAMPLE

Now take the famous German Credit Card data as an example. The data can be downloaded from the url https://onlinecourses.science.psu.edu/stat857/sites/onlinecourses.science.psu.edu.stat857/files/german_credit.csv

For the continuous variable, since its WOE must be monotonous increase or decrease, so I fit a linear regression model, take WOE as x variable, group number (1 2 3 4 …) as y variable, use genetic algorithm to make the model's square sum of residual minimum. Once square sum of residual is near zero, I can say WOE is monotonous increase or decrease. The following code is for the continuous variable, take variable 'Credit Amount' as an example.

Note: if the continuous variable was a ratio which is between 0 and 1, then need to multiply 100 or 1000 to let this code work.

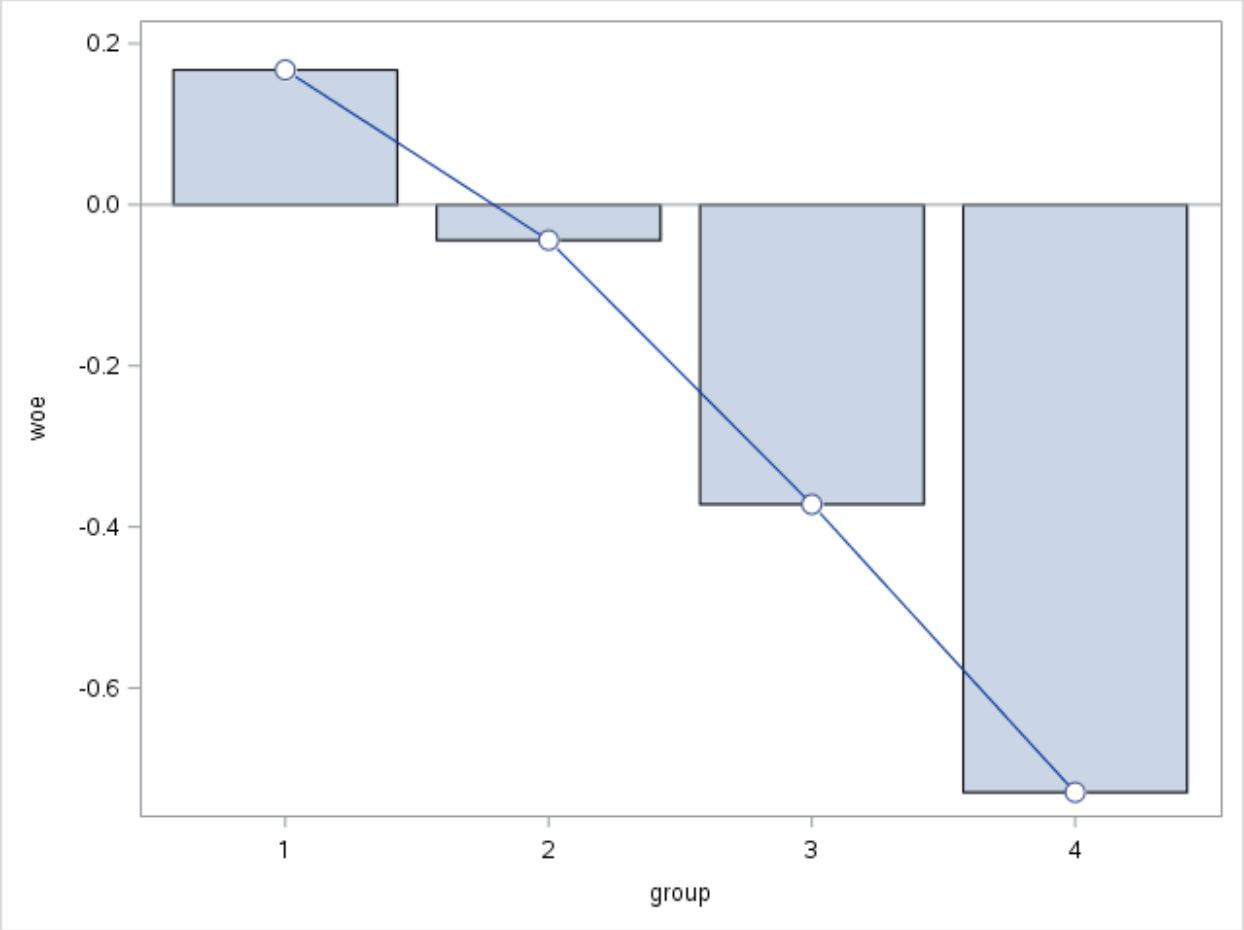
```
/* Download data from
https://onlinecourses.science.psu.edu/stat857/sites/onlinecourses.science.p
su.edu.stat857/files/german_credit.csv
*/

options validvarname=v7;
filename x '/folders/myfolders/german_credit.csv';
proc import datafile=x out=have dbms=csv replace;
run;



%let var=Credit_Amount;
%let group=4;
%let n_iter=100;
data temp;
 set have;
 good_bad=ifc(Creditability=1,'bad ','good');
 keep &var good_bad ;
run;

proc sql noprint;
 select sum(good_bad='bad'),sum(good_bad='good'),
        floor(min(&var)),ceil(max(&var)) into : n_bad,: n_good,: min,: max
  from temp;
quit;
%put &n_bad &n_good &min &max;
proc sort data=temp;by &var ;run;
proc iml;
use temp(where=(&var is not missing));
read all var {&var good_bad};
```

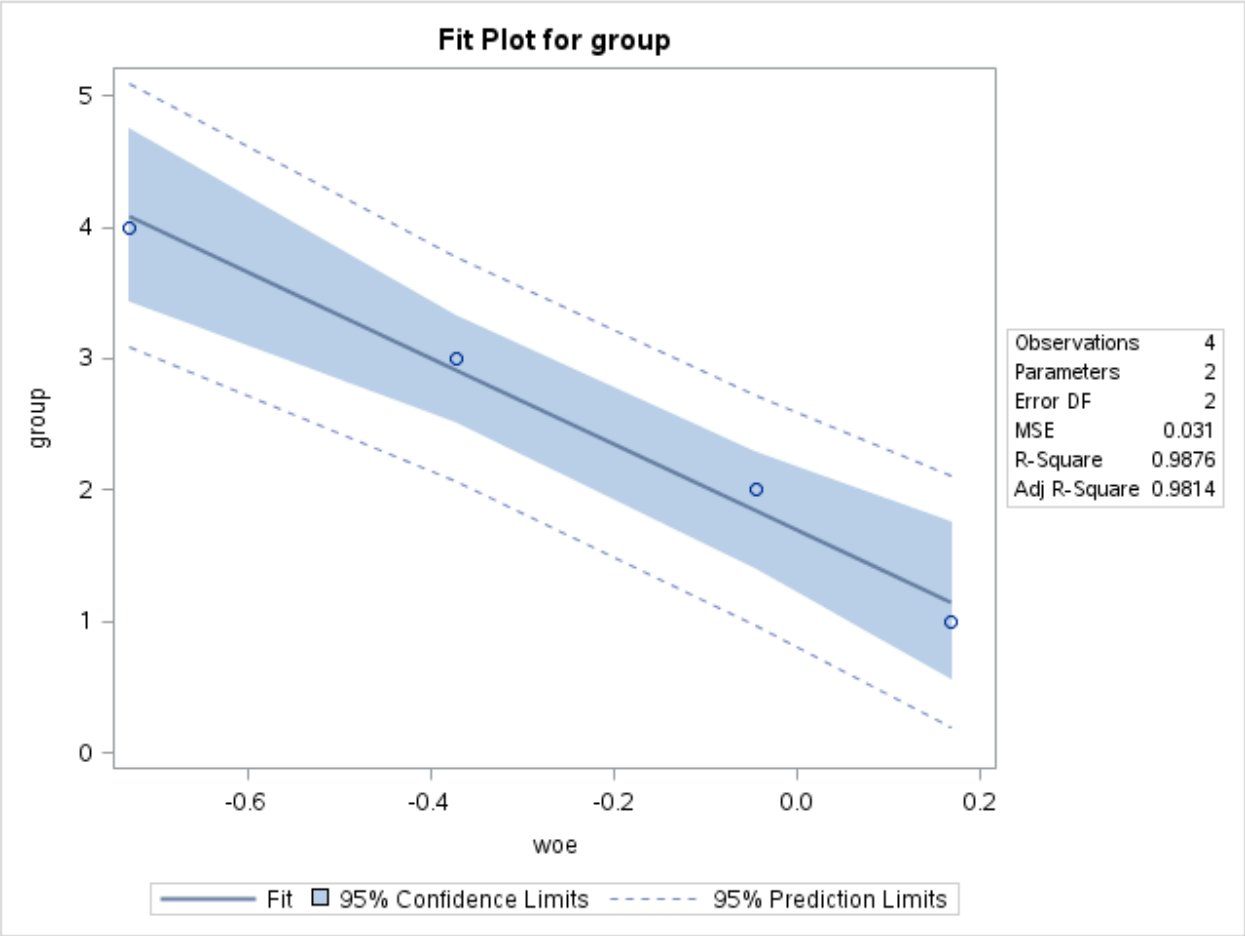
```
close;
start function(x) global(bin,&var ,good_bad,group,woe);
if countunique(x)=group-1 then do;
col_x=t(x);
call sort(col_x,1);
cutpoints= .M//col_x//.I ;
b=bin(&var ,cutpoints,'right');
if countunique(b)=group  then do;
do i=1 to group;
 idx=loc(b=i);
 temp=good_bad[idx];
 n_bad=sum(temp='bad');
 n_good=sum(temp='good');
 bad_dist=n_bad/&n_bad ;
 good_dist=n_good/&n_good ;
 if Bad_Dist>0.05 & Good_Dist>0.05  then woe[i]=log(Bad_Dist/Good_Dist);
  else woe[i]=.;
end;
if countmiss(woe)=0 then do;
xx=j(group,1,1)||woe;
beta=solve(xx`*xx,xx`*bin);
yhat=xx*beta;
sse=ssq(bin-yhat);
end;
else sse=999999;
end;
else sse=999999;
end;
else sse=999999;
return (sse);
finish;
group=&group ;
bin=t(1:group);
woe=j(group,1,.);

encoding=j(2,group-1,&min );
encoding[2,]=&max ;
id=gasetup(2,group-1,123456789);
call gasetobj(id,0,"function");
call gasetsel(id,10,1,1);
call gainit(id,1000,encoding);

niter = &n_iter;
do i = 1 to niter;
 call garegen(id);
 call gagetval(value, id);
end;
call gagetmem(mem, value, id, 1);
col_mem=t(mem);
call sort(col_mem,1);
cutpoints= .M//col_mem//.I ;
b=bin(&var ,cutpoints,'right');
create cutpoints var {cutpoints};
append;
close;
create group var {b};
append;
```

```
close;
print value[l = "Min Value:"] ;
call gaend(id);
quit;


data all_group;
 set temp(keep=&var rename=(&var=b) where=(b is missing)) group;
run;
data all;
 merge all_group temp;
 rename b=group;
run;


proc sql;
create table woe_&var as
 select group label=' ',count(*) as total label='total number',
min(&var) as min label='min value',max(&var) as max label='max value',
sum(good_bad='bad') as n_bad label='bad number',sum(good_bad='good') as
n_good label='good number',
sum(good_bad='bad')/(select sum(good_bad='bad') from all ) as bad_dist
label='bad percent',
sum(good_bad='good')/(select sum(good_bad='good') from all ) as good_dist
label='good percent',
log(calculated Bad_Dist/calculated Good_Dist) as woe
from all
   group by group
    order by woe;

select *,sum(  (Bad_Dist-Good_Dist)*woe  ) as iv
 from woe_&var ;
quit;

proc sgplot data=woe_&var noautolegend;
 vbar group/response=woe nostatlabel missing;
 vline group/response=woe nostatlabel missing markers
  MARKERATTRS=(symbol=circlefilled size=12)
  MARKERFILLATTRS=(color=white)
  MARKEROUTLINEATTRS=graphdata1
  FILLEDOUTLINEDMARKERS;
run;
ods select fitplot;
proc reg data=woe_&var;
model group=woe/ cli clm ;
quit;
```

Min Value:
0.0620377

group	total number	min value	max value	bad number	good number	bad percent	good percent	woe	iv
4	102	7166	18424	54	48	0.077143	0.16	-0.72951	0.088446
3	60	5771	7127	37	23	0.052857	0.076667	-0.37187	0.088446
2	139	3590	5743	96	43	0.137143	0.143333	-0.04415	0.088446
1	699	250	3578	513	186	0.732857	0.62	0.167231	0.088446





Output 1. Output from Code

The minimal value is near zero. And from the graph, there is apparently monotonous decrease relationship.

For the category variable, there is no requirement for WOE to be monotonous increase or decrease, but should get better IV after grouping variable. So I am solving an optimization problem (i.e. Make the IV largest) by genetic algorithm. The following code is for the category variable, take variable 'Purpose' as an example.

```
/* Download data from
https://onlinecourses.science.psu.edu/stat857/sites/onlinecourses.science.p
su.edu.stat857/files/german_credit.csv
*/

options validvarname=v7;
filename x '/folders/myfolders/german_credit.csv';
proc import datafile=x out=have dbms=csv replace;
run;
```

```
%let var=Purpose;
%let group=6;
%let n_iter=100;
data temp;
 set have;
 good_bad=ifc(Creditability=1,'bad ','good');
 keep &var good_bad ;
run;

proc sql noprint;
 select sum(good_bad='bad'),sum(good_bad='good') into : n_bad,: n_good
  from temp;
  create index &var on temp;
quit;
%put &n_bad &n_good ;
proc iml;
use temp ;
read all var {&var good_bad};
close;
start function(x) global(bin,&var ,good_bad,group,level,woe);
if countunique(x)=group then do;
do i=1 to group;
 idx=loc(x=i);
 levels=level[idx];
 index=loc(element(&var,levels));
 temp=good_bad[index];
 n_bad=sum(temp='bad');
 n_good=sum(temp='good');
 bad_dist=n_bad/&n_bad ;
 good_dist=n_good/&n_good ;
 if Bad_Dist>0.05 & Good_Dist>0.05 then woe[i]=(Bad_Dist-
Good_Dist)#log(Bad_Dist/Good_Dist);
  else woe[i]=.;
end;
if countmiss(woe)=0 then iv=sum(woe) ;

else iv=-999999;
end;
else iv=-999999;
return (iv);
finish;
group=&group ;
bin=t(1:group);
woe=j(group,1,.);

level=unique(&var);
n_level=countunique(&var);
encoding=j(2,n_level,1 );
encoding[2,]=group ;
id=gasetup(2,n_level,123456789);
call gasetobj(id,1,"function");
call gasetsel(id,10,1,1);
call gainit(id,1000,encoding);

niter = &n_iter ;
do i = 1 to niter;
```

```
 call garegen(id);
 call gagetval(value, id);
end;
call gagetmem(mem, value, id, 1);
col_mem=t(mem);
create group var{col_mem level};
append;
close;
print value[l = "IV Max Value:"] ;
call gaend(id);
quit;
data all;
 merge temp group(rename=(level=&var col_mem=group)) ;
 by &var;
run;

proc sql;
create table woe_&var as
 select group,count(*) as total label='total number',
sum(good_bad='bad') as n_bad label='bad number',sum(good_bad='good') as
n_good label='good number',
sum(good_bad='bad')/(select sum(good_bad='bad') from all ) as bad_dist
label='bad percent',
sum(good_bad='good')/(select sum(good_bad='good') from all ) as good_dist
label='good percent',
log(calculated Bad_Dist/calculated Good_Dist) as woe
from all
   group by group
    order by woe;

create index group on woe_&var;
create index col_mem on group;

select *,sum(  (Bad_Dist-Good_Dist)*woe  ) as iv
 from woe_&var ;
quit;
```

IV Max Value:
0.1661173

group	total number	bad number	good number	bad percent	good percent	woe	iv
3	84	49	35	0.07	0.116667	-0.51083	0.166117
5	234	145	89	0.207143	0.296667	-0.3592	0.166117
2	97	63	34	0.09	0.113333	-0.23052	0.166117
6	193	131	62	0.187143	0.206667	-0.09923	0.166117
1	280	218	62	0.311429	0.206667	0.410063	0.166117
4	112	94	18	0.134286	0.06	0.805625	0.166117

Output 2. Output from Code

REFERENCES

SAS Institution Inc.2014. *SAS/IML® 13.2 User's Guide*. Cary, NC: SAS Institute Inc.

Siddiqi, Naeem. 2006. Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring. John Wiley & Sons, Inc. Hoboken, NJ.

CONTACT INFORMATION

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

Xia Ke Shan
iFRE Inc.
Beijing, China
Phone: +8613521225927
E-mail: 12135835@qq.com

Matthew Kastin
NORC at the University of Chicago
Hummelstown, Pennsylvania
E-mail: fried.egg@VERIZON.NET

Peter Eberhardt
Fernwood Consulting Group Inc.
Toronto, Canada
E-mail: peter@fernwood.ca