# 1 Code MATLAB for MWS

### MaxScoreCompute.m

```
function [value,estimates,time,quality]=MaxScoreCompute()

%Main Program: Computes Max score by defining a MILP and calling milp.m
%tic
[X,y,w]=readXyw();
[X,mu,sigma]=standardizeX(X);
[c,A,b]=definecAb(X,y,w);
[lb,ub, Aeq, beq, n, best]=definelbub(X);
[x,score,feasible, time]=milp_cplex(c,A,b,Aeq,beq,lb,ub,n, best);
estimatesNorm=x
value=score
status=feasible
runtime=time
quality=status

estimatesRaw=denormalizeEstimates(estimatesNorm,mu,sigma)
estimates=estimatesRaw
%time=toc
% HILSquality=feasible;

end
```

### definecAb.m

```
function [c,A,b]=definecAb(X,y,w)

%Defines c,A,b for milp.m

n=size(X,1);
p=size(X,2);

c1=repmat(-1,1,n);
c2=repmat(0,1,p);
c=[c1 c2];

d=10;
%d=5;
for i=1:1:n
    M(i)=abs(X(i,1))+abs(X(i,2:end))*repmat(d,1,p-1)';
end
Abin=diag([M]);
for i=1:1:n
    for j=1:1:p
        Areal(i,j)=(1-2.*y(i)) * X(i,j) ;
```

```
    end
end
A=[Abin Areal];

b=M;

end
```

### definelbub.m

```
function [lb,ub, Aeq, beq, n, best]=definelbub(X)

%Defines lb,ub for milp.m
d=10;
%d=5;
n=size(X,1);
p=size(X,2);
lb1=repmat(0,1,n);
lb2=repmat(-d,1,p);
lb2(1)=1;
lb=[lb1 lb2];

ub1=repmat(1,1,n);
ub2=repmat(d,1,p);
ub2(1)=1;
ub=[ub1 ub2];

Aeq=[];
beq=[];

best=0;

end
```

### milp_cplex.m

```
function [x,score,feasible, time]=milp_cplex(c,A,b,Aeq,beq,lb,ub,n, best);
% Solves a mixed integer lp using gurobi 5.0.1
% c: is objective function coefficients A: is constraint matrix b: is constraint
    vector
% lb: lower bound ub: upper bound n: number of 0-1 variables
% best: is best solution so far
% Note this uses the MATLAB/Gurobi Interface documented at
% http://www.gurobi.com/documentation/5.0/reference-manual/node650
% Also, it assumes first n variables must be integer.
% The MIP equations of the maximum score estimator are available at
% Florios. K, Skouras, S. (2008) Exact computation of maximum weighted
% score estimators, Journal of Econometrics 146, 86-91.
```

```
% Written by Kostas Florios, July 20, 2012
%
% gurobi 5.0.1
% cd c:/Users/jones/gurobi500/win64/matlab
% gurobi_setup
%
% cd C:\gurobi501\win32\matlab
% gurobi_setup
%
% cplex 12.6
% addpath('C:\Program
    Files\IBM\ILOG\CPLEX_Studio_Preview126\cplex\matlab\x86_win32')
%
model.Aineq = sparse(A) ;
model.f = c ;
model.bineq = b ;
model.lb = lb ;
model.ub = ub ;
model.ctype = [repmat('B',size(b,2),1) ; repmat('C',size(c,2)-size(b,2),1)]' ;


opt = cplexoptimset('cplex') ;
opt.mip.display = 4 ;
opt.mip.interval = 1000 ;

opt.timelimit= 100 ;
opt.mip.tolerances.mipgap = 0.00 ;
opt.parallel = 1 ;
opt.threads = 4 ;
opt.mip.strategy.file = 3 ;
opt.workmem = 1024 ;
opt.emphasis.mip = 3 ;

opt.exportmodel = 'D:\storage\research\matlab2011\milp-cplex-2015\myModel.lp' ;

model.options = opt;


[x,fval,exitflag,output] = cplexmilp(model) ;

  fprintf('Optimization returned status: %s\n', output.message);
  fprintf('Objective Value: %e\n', fval);
  fprintf('(Wall clock) Time elapsed (s): %e\n', output.time);
  fprintf('Decision variables: the last ones (after the 1.00) are betas\n');
  %disp(x)


x=x((size(b,2)+1):size(c,2))
score=-fval;
feasible=output.message;
```

```
time=output.time;
return;
```

### readXyw.m

```
function [X,y,w]=readXyw()

%Reads X,y,w of given max score problem

X=load('X_Horowitz.txt');
%X=[X(:,2) X(:,3) X(:,4)];
X=X(:,2:end);
y=load('y_Horowitz.txt');
y=[y(:,2)];
w=load('w_Horowitz.txt');
w=[w(:,2)];


end
```

### denormalizeEstimates.m

```
function [estimatesRaw]=denormalizeEstimates(estimatesNorm,mu,sigma);

%denormalized estimatesNorm obtained by Gurobi MIP to estimatesRaw, which
%are meaningful to the user

%quick and dirty implementation, based on GAMS and Fortran Analogues
p=size(estimatesNorm,1);
betaNorm=estimatesNorm;

for j=1:1:p
    betaRaw(j)=0;
    betaHelp(j)=0;
end

for j=1:1:p
    if (sigma(j) ~=0)
        betaHelp(j)= betaNorm(j)./sigma(j);
    end
    if (sigma(j) ==0)
        for jj=1:1:p
            if (sigma(jj) ~=0)
                betaHelp(j)=betaHelp(j)-betaNorm(jj).*mu(jj)./sigma(jj) ;
            else
                jj0=jj;
            end
        end
```

```
        betaHelp(j)=betaHelp(j)+betaNorm(jj0);
    end
end

for j=1:1:p
   betaRaw(j)=betaHelp(j)./betaHelp(1);
end

estimatesRaw=betaRaw;

end
```

**standardizeX.m**

```
function [X,mu,sigma]=standardizeX(X)

%Standardizes X

mu=mean(X);
sigma=std(X);
testX=(X-repmat(mu,size(X,1),1)) ./ repmat(sigma,size(X,1),1);
p=size(X,2);
for j=1:1:p
if isnan(testX(:,j))
   X(:,j) = X(:,j);
else
    X(:,j) = testX(:,j);
end

end
```

# 2 Contained functions

1. **MaxScoreCompute.m** → main
2. **definecAb.m** → defines matrices c,A,b
3. **definelbub.m** → defines bounds lb,ub and matrices Aeq,beq
4. **denormalizeEstimates.m** → post-processing
5. **milp_cplex.m** → calls cplexmilp commercial function
6. **readXyw.m** → input from files X.txt and ys.txt
7. **standardizeX.m** → pre-processing

# 3 Guidelines for usage

We present the source code in Matlab in order to compute the MWS (including Max Score) estimator exactly (when technically possible) using MIP following (Florios & Skouras, 2008). There are several options on which MIP solver to use in MATLAB. Here is one option - for MATLAB ® versions before R2014a:

- the cplexmilp.m solver by IBM ILOG CPLEX OPTIMIZATION STUDIO 12.6 ® which is formatted in the milp_cplex() function we wrote, and is the state-of-the-art solver for N $\approx$ 500

For users of versions after R2014a (including R2014a):

- the intlinprog() solver by MATHWORKS ®, can be used for medium scale problems (N=200-400 observations)

Typical values for p are p $\in$ {2,3,4,5,6}. Often CPLEX can tackle problems up to N $\approx$ 500 and p $\leq$ 5. [1].

# References

Florios, K., & Skouras, S. (2008). Exact computation of max weighted score estimators. *Journal of Econometrics*, *146*(1), 86-91.

[1]It is possible to choose the DGPs in such a way that problems with N $\approx$ 800 and p = 5 can be solved