water

MDPI

Supplementary Information

A Simple Time-Varying Sensitivity Analysis (TVSA) for Assessment of Temporal Variability of Hydrological Processes

Yelena Medina[1, 2, *] and Enrique Muñoz [1, 2]

[1] Department of Civil Engineering, Universidad Católica de la Santísima Concepción, Concepción 4090541, Chile; emunozo@ucsc.cl

[2] Centro de Investigación en Biodiversidad y Ambientes Sustentables CIBAS, Concepción 4090541, Chile

* Correspondence: ypmedina@ing.ucsc.cl; Tel.: +56-41-2345355

Received: 25 June 2020; Accepted: 31 August 2020; Published: 2 September 2020

```
function TVSA(datei,pars,sim,obs,strpars)
% ------------------------------------------------------------- %
% This function performs Time-Varying Sensitivity Analysis based on
% Regional Sensitivity Analysis and plots the sensitivity index and
% parameter values.
% The index is the Maximum Vertical Distance (MVD) between the two
% CDFs (behavioral and non-behavioral).
% ------------------------------------------------------------- %
%  datei  :    initial date - format                [yyyy,mm,dd]
%  pars   :    parameter matrix                     [N,P]
%  sim    :    ouput time series matrix             [T,N]
%  obs    :    observed time series vector          [T,1]
%  strpars:    cell with parameter names            {1,P} {'A','B',...}
% ------------------------------------------------------------- %
% N: sample size (simulations)
% P: number of parameters
% T: size of the time series
% ------------------------------------------------------------- %

%% Check inputs
if ~isnumeric(datei);error('input ''datei'' must be a vector of size
(1,3)');end
if ~isnumeric(pars);error('input ''pars'' must be a matrix of size
(N,P)');end
if ~isnumeric(sim);error('input ''sim'' must be a matrix of size (T,N)');end
if ~isnumeric(obs);error('input ''obs'' must be a vector of size (T,1)');end
if ~iscell(strpars);error('input ''strpars'' must be a cell array of size
{1,P}');end
[N,P]=size(pars);[T,Nn]=size(sim);[Tt,x]=size(obs);[x,Pp]=size(strpars);
if T~=Tt;error('input ''sim'' and ''obs'' must have the same number of
rows');end
if N~=Nn;error('Columns of input ''sim'' must be equal to rows of input
''pars''');end
if P~=Pp;error('input ''pars'' and ''strpars'' must have the same number of
columns');end
```

```
%% Input Window
d = dialog('Position',[500 300 250 200],'Name','Input Window');
txt1 = uicontrol('Parent',d,'Style','text','Position',[10 170 80
20],'String','Window Step :');
popup1 = uicontrol('Parent',d,'Style','popupmenu','Position',[15 150 80
20],'String',{'Daily';'Monthly';'Annual'});
txt2 = uicontrol('Parent',d,'Style','text','Position',[10 115 80
20],'String','WSize(2n+1) n:');
edit1 = uicontrol('Parent',d,'Style','edit','Position',[15 95 80
20],'String','1');
txt3 = uicontrol('Parent',d,'Style','text','Position',[140 170 80
20],'String','Criteria    :');
popup2 = uicontrol('Parent',d,'Style','popupmenu','Position',[155 150 80
20],'String',{'KGE';'NSE';'RMSE';'NSEiQ'});
txt4 = uicontrol('Parent',d,'Style','text','Position',[142 115 80
20],'String','Threshold :');
edit2 = uicontrol('Parent',d,'Style','edit','Position',[155 95 80
20],'String','0');
btn1 = uicontrol('Parent',d,'Style','pushbutton','Position',[40 40 70
25],'String','Ok','Callback',@btn1_Callback);
btn2 = uicontrol('Parent',d,'Position',[140 40 70
25],'String','Close','Callback','delete(gcf)');

%% Time-Varying Sensitivity Analysis
    function btn1_Callback(hObject, eventdata)
        windowstep        = popup1.Value;
        window            = str2num(get(edit1,'string'));
        ObjectiveFunction = popup2.Value;
        threshold         = str2num(get(edit2,'string'));
        close(gcf)
%%%%%%%%%%%%%%%% Check input window %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        if ~isnumeric(threshold)
            error('input ''threshold'' must be a number')
        end
        if ~isnumeric(window) || window<1 || window~=round(window)
            error('input ''window'' must be real positive integer')
        end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        for i=1:size(obs,1)
            datev(i,1)=datetime(datei(1),datei(2),datei(3)+i-1);
        end
        dateLb=datenum(datev);
        switch windowstep
            case 1
                t=size(sim,1);
                Temp='Days';
            case 2
                mnths=unique([year(datev) month(datev)],'rows');
                t=length(mnths);
                Temp='Months';
            case 3
                yrs=unique([year(datev)],'rows');
                t=length(yrs);
                Temp='Years';
        end
%%%%%%%%%%%%%%%% Objective Function %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        h_w=waitbar(0,'Running Time-Varying Sensitivity Analysis Algorithm');
```

```
        for dt=1:t
            for no=1:size(sim,2)
                if dt<window+1
                    switch windowstep
                        case 1 % Daily

step=find(datev==datetime(datei(1),datei(2),datei(3)+dt+window-1));
                        case 2 % Monthly
                            ms=datetime(datei(1),datei(2)+dt+window-1,1);

step=find(datev==datetime(datei(1),datei(2)+dt+window-
1,eomday(year(ms),month(ms))));
                        case 3 % Annual
                            yr=datetime(datei(1)+dt+window-1,datei(2)-1,1);
                            step=find(datev==datetime(datei(1)+dt+window-
1,datei(2)-1,eomday(year(yr),month(yr))));
                    end
                    obse=obs(1:step);
                    sim0=sim(1:step,no);
                    NN=size(sim0,1);
                    switch ObjectiveFunction
                        case 1 % KGE
                            r=corrcoef(sim0,obse,'rows','pairwise');
                            r=r(1,2);
                            alfag=sqrt(nanvar(sim0)/nanvar(obse));
                            betag=nanmean(sim0)/nanmean(obse);
                            ED=sqrt((r-1)^2+(alfag-1)^2+(betag-1)^2);
                            OF(no,dt)=1-ED;
                        case 2 % NSE
                            OF(no,dt)=1-(nansum((obse-
sim0).^2))/(nansum((obse-nanmean(obse)).^2));
                        case 3 % RMSE
                            OF(no,dt)=sqrt(1/NN*nansum((sim0-obse).^2));
                        case 4 % NSEiQ
                            for i=1:length(obse)
                                if obse(i)==0
                                    e=nanmean(obse)/100;
                                    break
                                else
                                    e=0;
                                end
                            end
                            OF(no,dt)=1-(nansum((1/(obse+e)-
1/(sim0+e)).^2))/(nansum((1/(obse+e)-nanmean(1/(obse+e))).^2));
                    end
                elseif t-dt<window+1
                    switch windowstep
                        case 1 % Daily

step=find(datev==datetime(datei(1),datei(2),datei(3)+dt-window-1));
                        case 2 % Monthly
                            step=find(datev==datetime(datei(1),datei(2)+dt-
window-1,1));
                        case 3 % Annual
                            step=find(datev==datetime(datei(1)+dt-window-
1,datei(2),1));
                    end
```

```
                    obse=obs(step:size(sim,1));
                    sim0=sim(step:size(sim,1),no);
                    NN=size(sim0,1);
                    switch ObjectiveFunction
                        case 1 % KGE
                            r=corrcoef(sim0,obse,'rows','pairwise');
                            r=r(1,2);
                            alfag=sqrt(nanvar(sim0)/nanvar(obse));
                            betag=nanmean(sim0)/nanmean(obse);
                            ED=sqrt((r-1)^2+(alfag-1)^2+(betag-1)^2);
                            OF(no,dt)=1-ED;
                        case 2 % NSE
                            OF(no,dt)=1-(nansum((obse-
sim0).^2))/(nansum((obse-nanmean(obse)).^2));
                        case 3 % RMSE
                            OF(no,dt)=sqrt(1/NN*nansum((sim0-obse).^2));
                        case 4 % NSEiQ
                            for i=1:length(obse)
                                if obse==0
                                    e=0.3;
                                    break
                                else
                                    e=0;
                                end
                            end
                            OF(no,dt)=1-(nansum((1/(obse+e)-
1/(sim0+e)).^2))/(nansum((1/(obse+e)-nanmean(1/(obse+e))).^2));
                    end
                else
                    switch windowstep
                        case 1 % Daily

stepa=find(datev==datetime(datei(1),datei(2),datei(3)+dt+window-1));

stepb=find(datev==datetime(datei(1),datei(2),datei(3)+dt-window-1));
                        case 2 % Monthly
                            msa=datetime(datei(1),datei(2)+dt+window-1,1);

stepa=find(datev==datetime(datei(1),datei(2)+dt+window-
1,eomday(year(msa),month(msa))));
                            stepb=find(datev==datetime(datei(1),datei(2)+dt-
window-1,1));
                        case 3 % Annual
                            yra=datetime(datei(1)+dt+window-1,datei(2)-1,1);
                            stepa=find(datev==datetime(datei(1)+dt+window-
1,datei(2)-1,eomday(year(yra),month(yra))));
                            stepb=find(datev==datetime(datei(1)+dt-window-
1,datei(2),1));
                    end
                    obse=obs(stepb:stepa);
                    sim0=sim(stepb:stepa,no);
                    NN=size(sim0,1);
                    switch ObjectiveFunction
                        case 1 % KGE
                            r=corrcoef(sim0,obse,'rows','pairwise');
                            r=r(1,2);
                            alfag=sqrt(nanvar(sim0)/nanvar(obse));
```

```
                betag=nanmean(sim0)/nanmean(obse);
                ED=sqrt((r-1)^2+(alfag-1)^2+(betag-1)^2);
                OF(no,dt)=1-ED;
            case 2 % NSE
                OF(no,dt)=1-(nansum((obse-
sim0).^2))/(nansum((obse-nanmean(obse)).^2));
            case 3 % RMSE
                OF(no,dt)=sqrt(1/NN*nansum((sim0-obse).^2));
            case 4 % NSEiQ
                for i=1:length(obse)
                    if obse==0
                        e=0.3;
                        break
                    else
                        e=0;
                    end
                end
                OF(no,dt)=1-(nansum((1/(obse+e)-
1/(sim0+e)).^2))/(nansum((1/(obse+e)-nanmean(1/(obse+e))).^2));
            end
        end
    end
%%%%%%%%%%%%%%%%% Sensitivity Analysis %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Behavioral and Non-Behavioral Models
    L=OF(:,dt);
    [N,P]=size(L);
    for i=1:size(pars,2)
        Pop=pars(:,i);
        if ObjectiveFunction==2 % RMSE
            if threshold==0
                threshold=0.2*nanmean(obse);
            end
            thrs=sum(OF(:,dt)>repmat(threshold,N,1),2)==P;
        else % KGE - NSE - NSEiQ
            thrs=sum(OF(:,dt)<repmat(threshold,N,1),2)==P;
        end
        B=Pop(~thrs,1);% Behavioral Models (B-M)
        nB=size(B,1); % Number of B-M
        NB=Pop(thrs,1);% Non-Behavioral Models (NB-M)
        nNB=size(NB,1); % Number of NB-M
        if nB<=0 || nNB<=0
            mvd(:,dt)=0;
            parB(i,dt)=NaN;
            parNB(i,dt)=NaN;
        else % CDFs of B and NB models
            xx=unique(sort(Pop(:,1)));
            %%% B models %%%
            parB(i,dt)=prctile(B,50); % Parameter values of B models
            tmxB=sort(B);
            tmyB=(1:length(tmxB))/length(tmxB);
            [tmxB,iuB] = unique(tmxB,'last');
            tmyB=tmyB(iuB);
            FiB=ones(size(xx));
            for j=length(tmyB):-1:1
                FiB(xx<=tmxB(j))=tmyB(j);
            end
            FiB(xx<tmxB(1))=0;
```

```
                    %%% NB models %%%
                    parNB(i,dt)=prctile(NB,50); % Parameter values of NB
models
                    tmxNB=sort(NB);
                    tmyNB=(1:length(tmxNB))/length(tmxNB);
                    [tmxNB,iuNB] = unique(tmxNB,'last');
                    tmyNB=tmyNB(iuNB);
                    FiNB=ones(size(xx));
                    for jj=length(tmyNB):-1:1
                        FiNB(xx<=tmxNB(jj))=tmyNB(jj);
                    end
                    FiNB(xx<tmxNB(1))=0;
                    % Maximum Vertical Difference (MVD)
                    mvd(i,dt) = max(abs(FiB-FiNB));
                end
            end
            clear L
            waitbar(dt/t);
        end
        close(h_w);
%%%%%%%%%%%%%%%%%% PLOT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if nNB==0
    warndlg('Cannot find any value below (above-RMSE) the
threshold','¡Warning!')
elseif nB==0
    warndlg('Cannot find any value above (below-RMSE) the
threshold','¡Warning!')
else
    if  windowstep==1
        mvdf=mvd; parBf=parB; parNBf=parNB;
    end
    for i=1:t
        for f=1:size(obs,1)
            if  windowstep==2 & mnths(i,:)==[year(datev(f)) month(datev(f))]
                mvdf(:,f)=mvd(:,i);
                parBf(:,f)=parB(:,i);
                parNBf(:,f)=parNB(:,i);
            end
            if  windowstep==3 & yrs(i,:)==year(datev(f))
                mvdf(:,f)=mvd(:,i);
                parBf(:,f)=parB(:,i);
                parNBf(:,f)=parNB(:,i);
            end
        end
    end
    figure
    yyaxis left
    colrs=gray;
    colrs=colrs(end:-1:1,:);
    colormap(colrs)
    imagesc(dateLb,[1:size(pars,2)],mvdf); hold on
    ylabel('Parameters')
    set(gca,'YTick',1:size(pars,2),'YTickLabel',strpars)
    colorbar
    yyaxis right
    plot(dateLb,obs,':r','LineWidth',1)
    ylabel('Flow, m^3/s')
```

```
    datetick('x',28,'keeplimits')
    xlabel('Time')
    title(['Time-Varying Sensitivity Analysis - Window Size:
',num2str(2*window+1),' ', Temp])

    if size(pars,2)<=4
        subp='2,2,';
    elseif size(pars,2)>4&&size(pars,2)<=6
        subp='3,2,';
    elseif size(pars,2)>6&&size(pars,2)<=9
        subp='3,3,';
    elseif size(pars,2)>9&&size(pars,2)<=12
        subp='4,3,';
    else %size(pars,2)>12&&size(pars,2)<=16
        subp='4,4,';
    end

    figure
    for i=1:size(pars,2)
        if size(pars,2)>1
            eval(['subplot(' subp num2str(i) ')'])
        end
        scatter(dateLb,parBf(i,:),3,'b','filled');hold on;
        scatter(dateLb,parNBf(i,:),3,'r','filled');hold on;
        plot(dateLb,parBf(i,:),'-b','linewidth',1);hold on;
        plot(dateLb,parNBf(i,:),'-r','linewidth',1);hold on;
        datetick('x',28,'keeplimits')
        xlim([dateLb(1) dateLb(end)])
        ylabel(['' strpars{1,i} '']);
    end
end
    end
end
```