HW5 STAT5376

Dynamic programming with SRSF

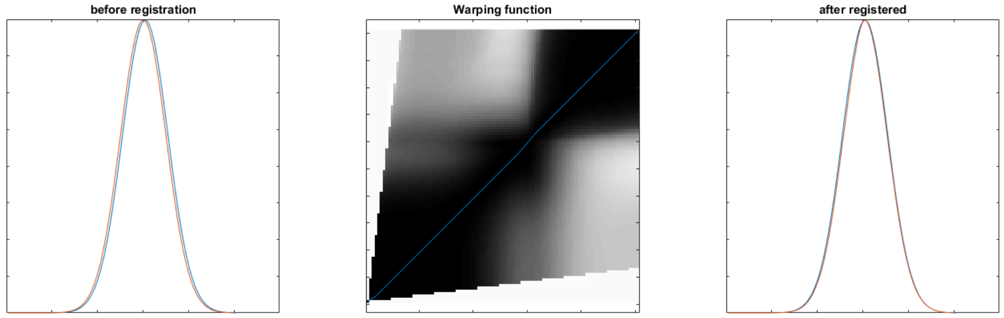
Li Sun

November 19, 2016

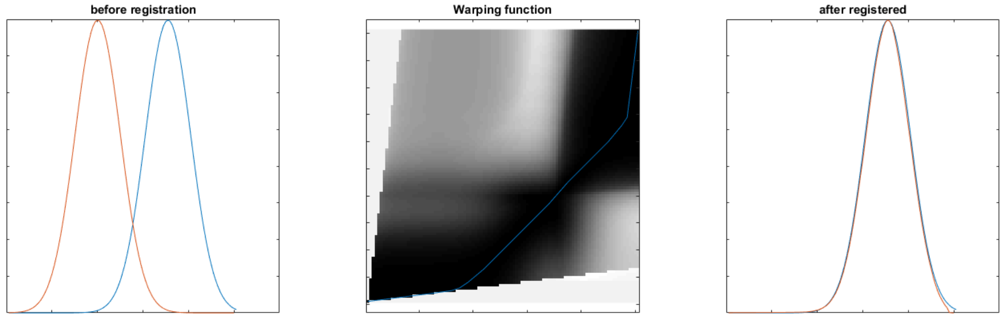
PART I: Smooth function registration

To demonstrate the dynamic programming with SRSF, I simulated several pairs of functions and try to register them.

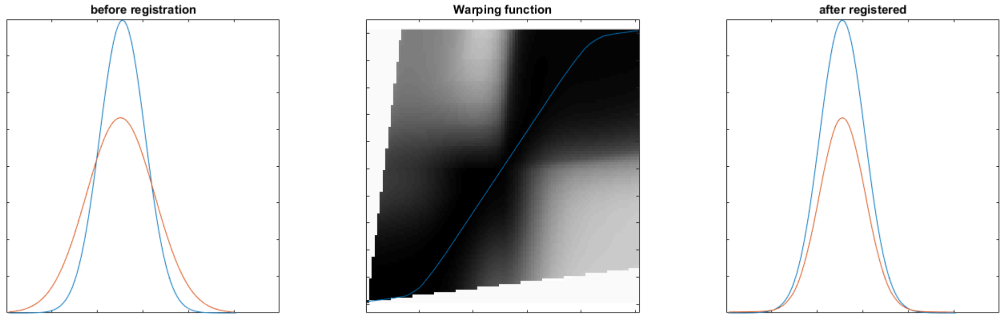
1.First to register two identical function.



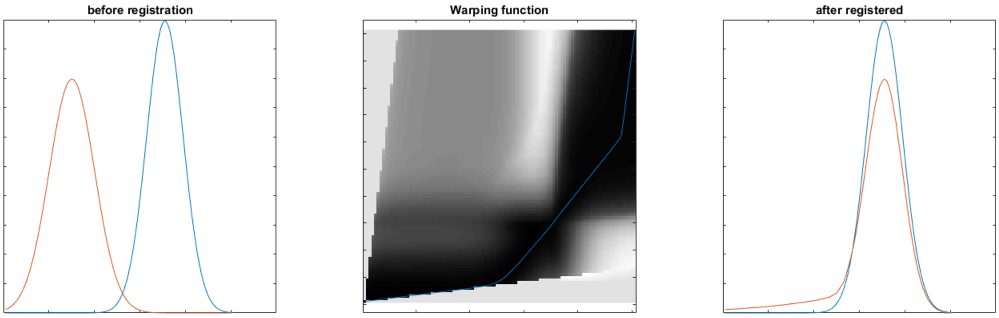
2.Register 2 functions with different locations but exact same shapes



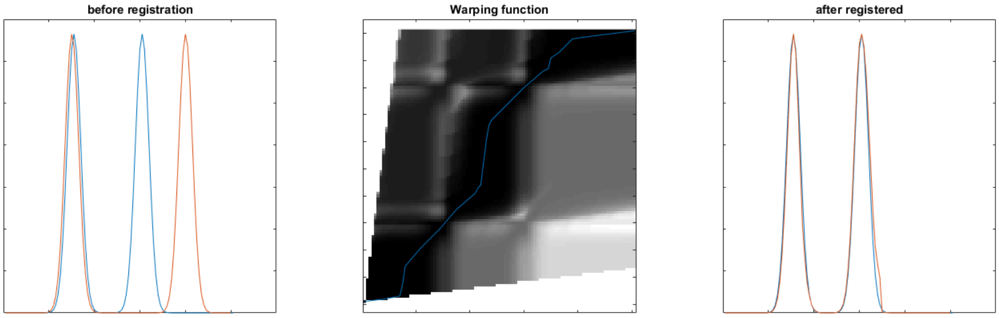
3.Functions with different variation but same location.



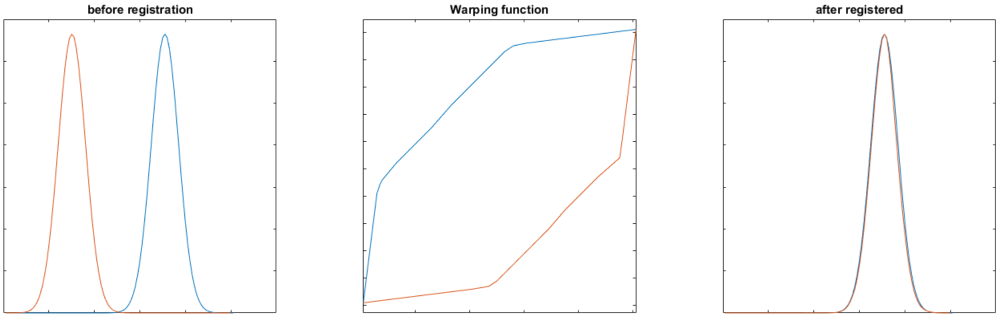
4.Functions with different center and different variations.



5.Let’s try warp bi-peak function.

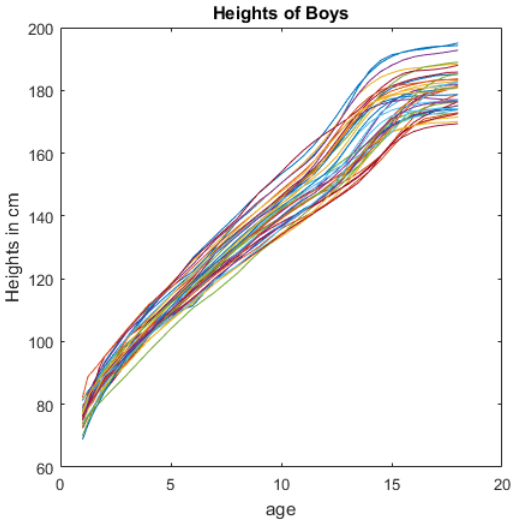


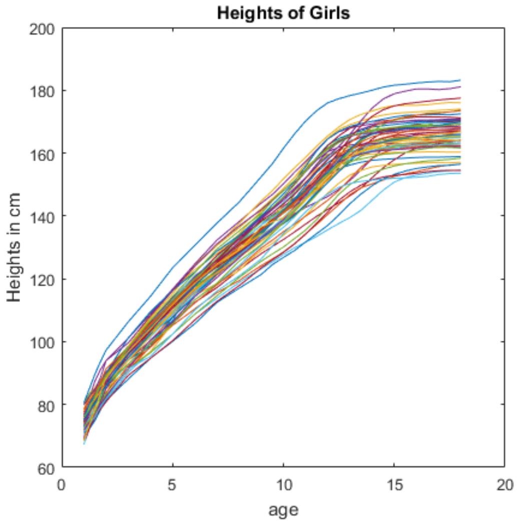
6.If warp g to f, can we observe inverted warping function?



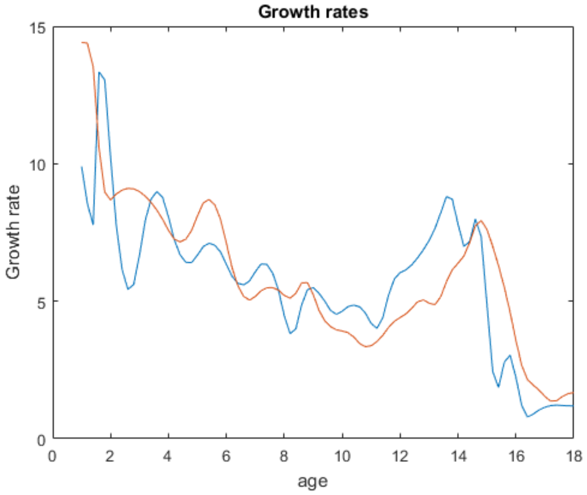
Above all, the algorithm searches 43 points as neighbors with 43 different slopes. It calculate fast and works well under most situations.Indeed, if we change the order of 2 functions to be warped, the warping function is inverse to each other.
However, I do observed some distortion when I tried to register a curve with larger magnitude to another curve with lower magnitude. The magnitude is well reserved which is an improvement comparing to L2-norm registration.

PART II: smooth and register growth data.
Data from R package fda as in following figure:

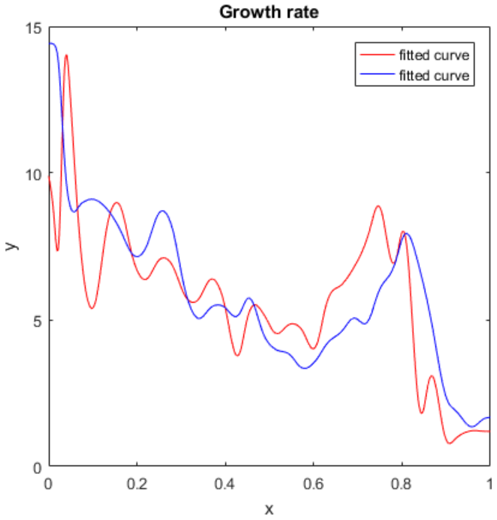


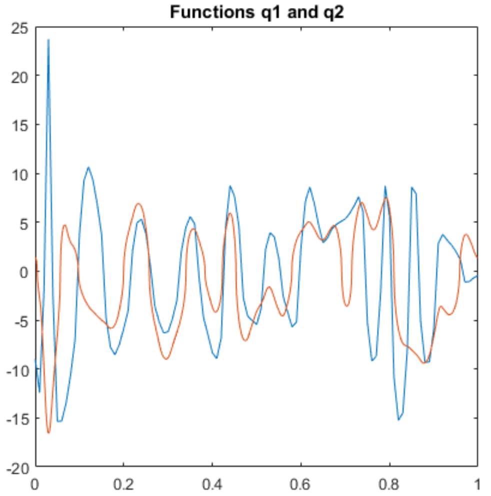


Because we are interested in the growth rates instead of absolute heights, so I first smooth the data and get a growth rate function of 2 random curves.

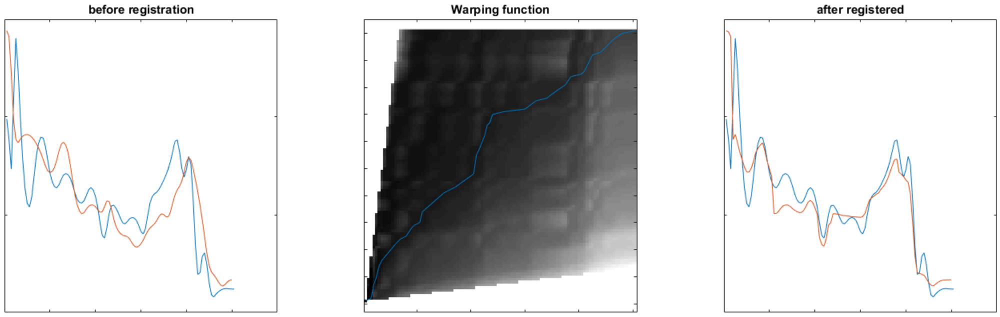


Next, I smoothed the growth rates curve and converted them to SRSF functions q1 and q2 which are ready for registration.

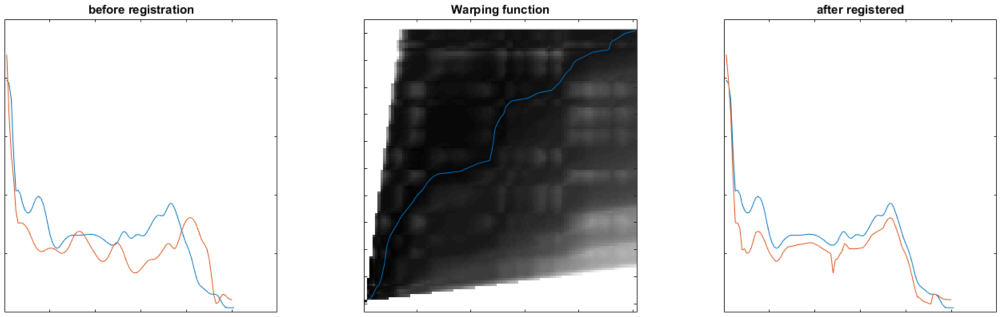




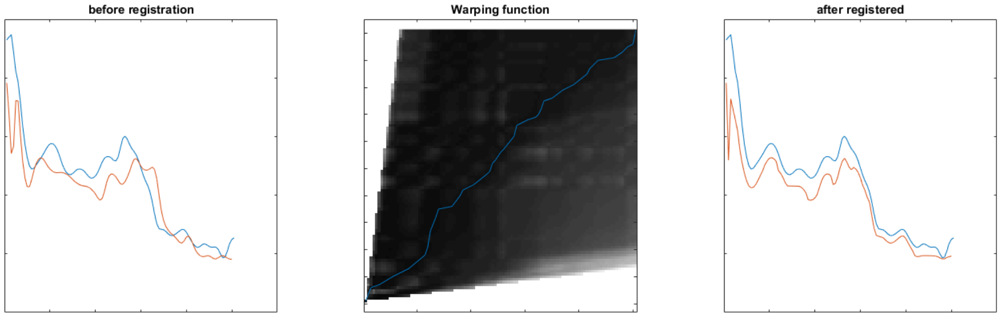
Now I applied dynamic programming with SRSF representations



Try another pair of boys



Try another pair of girls



Above all, the algorithm seem to work well for growth data.

All code please see https://github.com/rikku1983/STAT5376

Thanks!

```
%Part I
clear all;
close all;
%define functions
n=100;
m=1000;
xx1=(0:n)/n;
xx2=(0:m)/m;
f=normpdf(xx1,0.2,0.02)+normpdf(xx1,0.4,0.02)*0.6;
g=normpdf(xx2,0.6,0.02)+normpdf(xx2,0.8,0.02);

%Smooth x1 and x2 by splines
smthpara=1;
fs=fit(xx1', f', 'smoothingspline', 'SmoothingParam', smthpara);
gs=fit(xx2', g', 'smoothingspline', 'SmoothingParam', smthpara);
%Generate q1 from f and q2 from g

%x2=0:1/m:1;
for i = 1:length(xx1)
    q1(i)=sign((fs(xx1(i)+0.0001)-fs(xx1(i)-
0.0001))/(2*0.0001)).*sqrt(abs((fs(xx1(i)+0.0001)-fs(xx1(i)-
0.0001))/(2*0.0001)));
end
for i=1:length(xx2)
    q2(i)=sign((gs(xx2(i)+0.0001)-gs(xx2(i)-
0.0001))/(2*0.0001)).*sqrt(abs((gs(xx2(i)+0.0001)-gs(xx2(i)-
0.0001))/(2*0.0001)));
end

[path, E]=sldpSRSF2(q1,q2);

close all;
fig=figure();
set(fig,'Position', [200 200 1600 450]);
subplot(131);plot(f);hold on;plot(g((1:n)*m/n));title('before
registration');
set(gca,'XTickLabel',[],'YTickLabel',[]);
subplot(132);imagesc(E');colormap(gray);hold
on;plot(path(:,1),path(:,2));axis xy;
axis equal;set(gca,'XTickLabel',[],'YTickLabel',[]);
title('Warping function');
subplot(133);plot(f);hold on;
plot((1:n),g(round(interp1(path(:,1),path(:,2),1:n)*(m+1)/(n+1))));titl
e('after registered');
set(gca,'XTickLabel',[],'YTickLabel',[]);


%%%Part2, Growth data.
close all, clear all;
dat=csvread('bgd.csv',1,0);
age=dat(:,1);
boy=dat(:,2:40);
girl=dat(:,41:length(dat));
%Visualization of data
fig1=figure();
set(fig1,'Position', [200 200 1100 450]);
```

```
subplot(121);plot(age,boy);title('Heights of
Boys');xlabel('age');ylabel('Heights in cm');
subplot(122);plot(age,girl);title('Heights of
Girls');xlabel('age');ylabel('Heights in cm');
%Derivatives(Growth rates)
%Smooth function and find growth rates.
%Pick two curve.
c1=girl(:,29);
c2=girl(:,28);
smthpara=1;
c1s=fit(age,c1, 'smoothingspline', 'SmoothingParam', smthpara);
c2s=fit(age,c2, 'smoothingspline', 'SmoothingParam', smthpara);
%Generate new functions gr1 and gr2 for derivative
x=1:0.2:18;
for i = 1:length(x)
    gr1(i)=(c1s(x(i)+0.02)-c1s(x(i)-0.02))/(2*0.02);
end
for i=1:length(x)
    gr2(i)=(c2s(x(i)+0.02)-c2s(x(i)-0.02))/(2*0.02);
end
%plot the functions of growth rate to be registered
close all;
plot(x,gr1);hold on;plot(x,gr2);title('Growth
rates');xlabel('age');ylabel('Growth rate');

%Now lets calculate Q
%Smooth x1 and x2 by splines and scale the curve to between 0 and 1
smthpara=1;
fs=fit(((x-1)/max(x-1))', gr1', 'smoothingspline', 'SmoothingParam',
smthpara);
gs=fit(((x-1)/max(x-1))', gr2', 'smoothingspline', 'SmoothingParam',
smthpara);
close all;
fig1=figure();
set(fig1,'Position', [200 200 1100 450]);
subplot(121);plot(fs);hold on; plot(gs, 'b');title('Growth rate');
%Generate q1 from f and q2 from g
%x2=0:1/m:1;
n=100;
m=1000;
xx1=(0:n)/n;
xx2=(0:m)/m;

for i = 1:length(xx1)
    q1(i)=sign((fs(xx1(i)+0.0001)-fs(xx1(i)-
0.0001))/(2*0.0001)).*sqrt(abs((fs(xx1(i)+0.0001)-fs(xx1(i)-
0.0001))/(2*0.0001)));
end
for i=1:length(xx2)
    q2(i)=sign((gs(xx2(i)+0.0001)-gs(xx2(i)-
0.0001))/(2*0.0001)).*sqrt(abs((gs(xx2(i)+0.0001)-gs(xx2(i)-
0.0001))/(2*0.0001)));
end
subplot(122);plot(xx1,q1);hold on; plot(xx2,q2);title('Functions q1 and
q2');
```

```
% Ready to register
[path, E]=sldpSRSF2(q1,q2);

close all;
f=fs(xx1);
g=gs(xx2);
fig=figure();
set(fig,'Position', [200 200 1600 450]);
subplot(131);plot(f);hold on;plot(g((1:n)*m/n));title('before
registration');
set(gca,'XTickLabel',[],'YTickLabel',[]);
subplot(132);imagesc(E');colormap(gray);hold
on;plot(path(:,1),path(:,2));axis xy;
axis equal;set(gca,'XTickLabel',[],'YTickLabel',[]);
title('Warping function');
subplot(133);plot(f);hold on;
plot((1:n),g(round(interp1(path(:,1),path(:,2),1:n)*(m+1)/(n+1))));titl
e('after registered');
set(gca,'XTickLabel',[],'YTickLabel',[]);
```

```
%f,g: 2 functions
%n: points of f
%m: points of g
%This function search neiborghood v
function [path,E]=sldp(f,g)
c=inf;
n=length(f);
m=length(g);
xx1=(0:n-1)/(n-1);
xx2=(0:m-1)/(m-1);
E=zeros(n,n);
E(1,:)=c;
E(:,1)=c;
E(1,1)=0;
v=[1,1;2,1;3,1;4,1;5,1;6,1;1,2;1,3;1,4;1,5;1,6;2,3;3,2;3,4;4,3;2,5;3,5;
4,5;5,2;5,3;5,4;5,6;6,5;

1,7;2,7;3,7;4,7;5,7;6,7;7,1;7,2;7,3;7,4;7,5;7,6;1,8;3,8;5,8;7,8;8,7;8,5
;8,3;8,1];
for i=2:n;
    for j=2:n;
        for r=1:size(v,1);
            k=i-v(r,1);
            l=j-v(r,2);
            if (k>0 && l>0)
                CandE(r) = E(k,l) + energySRSF2(f,g,k,l,i,j);
            else
                CandE(r)=c;
            end
        end
        [E(i,j),idx] =min(CandE);
        path(i,j,1) = i-v(idx,1);
        path(i,j,2) = j-v(idx,2);
    end
```

```
end
%reconstruct gamma
x(1) = n;
y(1) = n;
cnt = 1;
while x(cnt)>1;
    x(cnt+1) = path(x(cnt),y(cnt),1);
    y(cnt+1) = path(x(cnt),y(cnt),2);
    cnt = cnt+1;
end
path=[x',y'];
```

```
function cost=energy(f,g,k,l,i,j)
%This is the function to calculate 2-norm distence between two fuction
%between path from (k,l) to (i,j) used in dynamic programming
n=length(f);
m=length(g);
slope=(j-l)/(i-k);
gidx=round((l+((k+1:i)-k).*slope)/n*m);
cost=norm(f(k+1:i)-g(gidx)*sqrt(slope))^2;
```