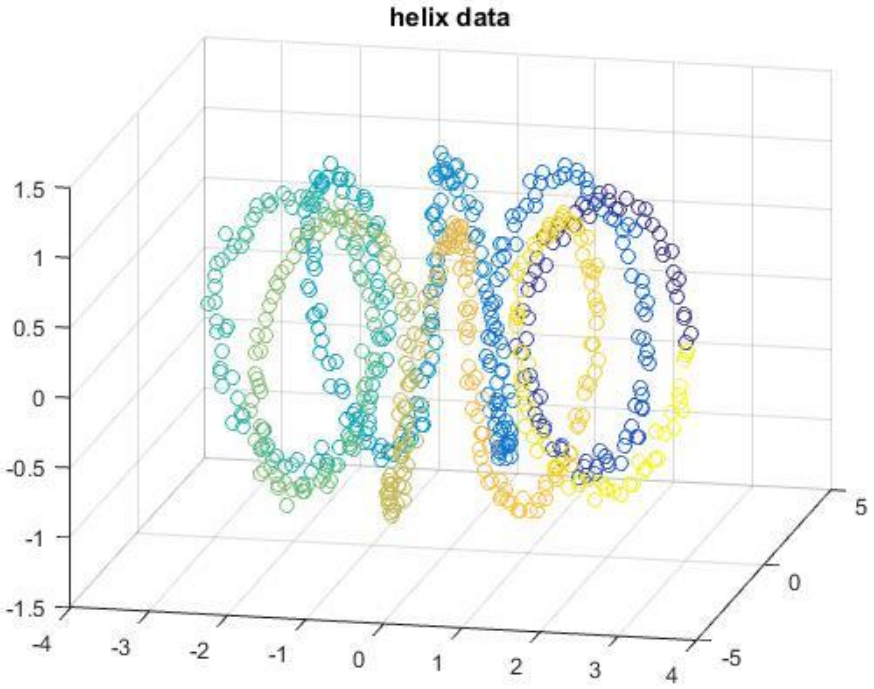
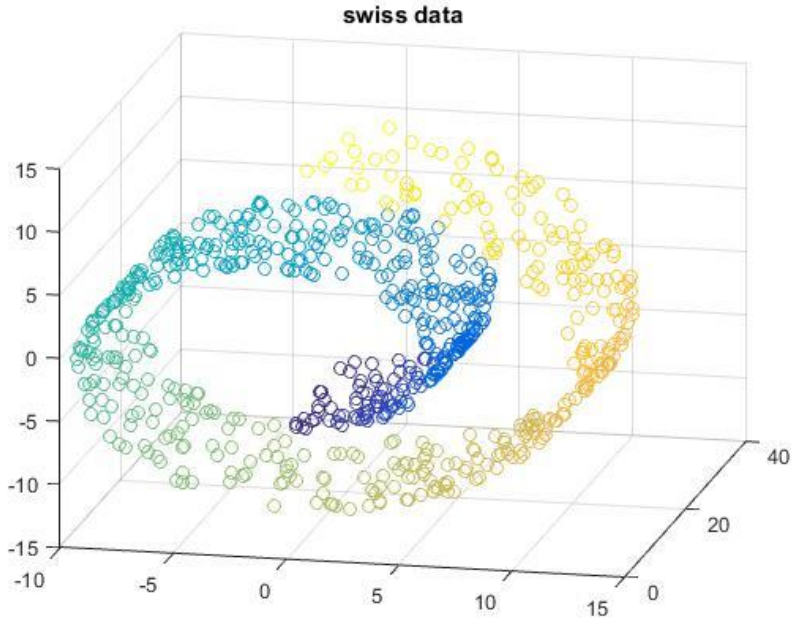
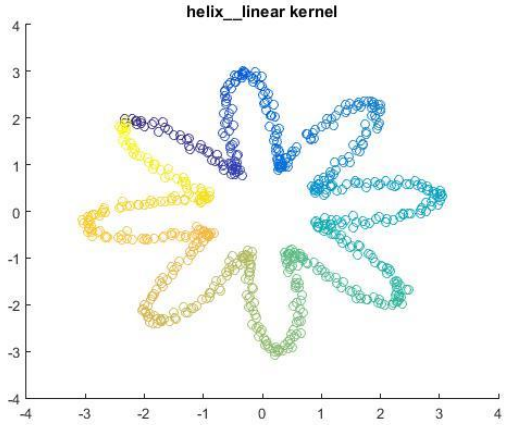
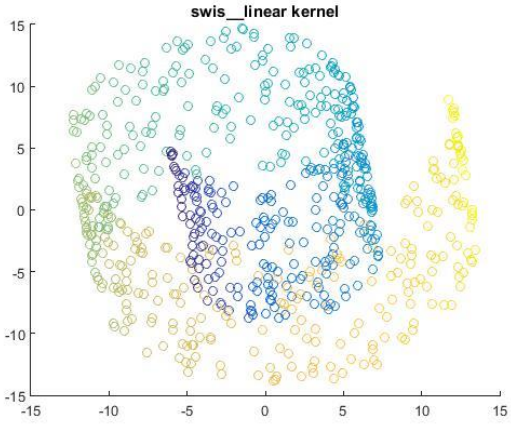
Part a

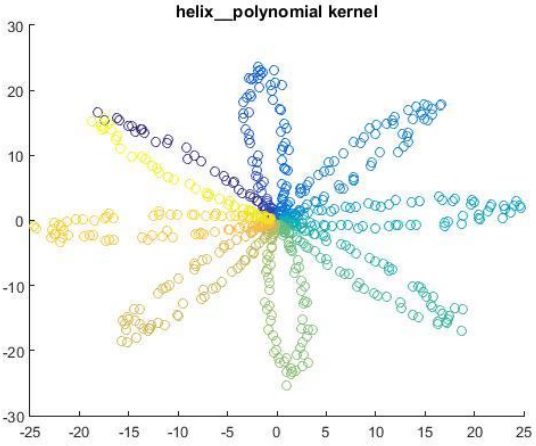

helix data
1.5
1
0.5
0
-0.5
-1
-1.5
-4
-3
-2
-1
0
1
2
3
4
5
0
-5

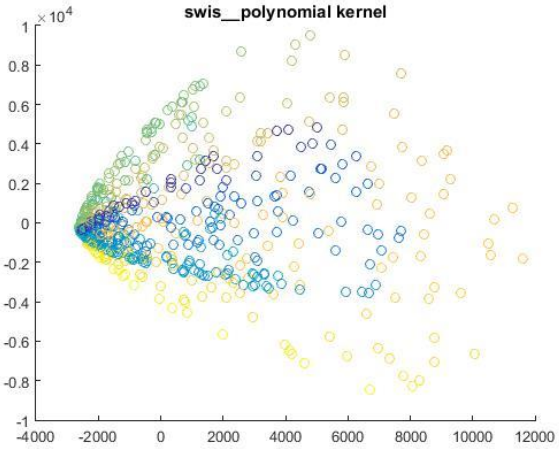

swiss data
15
10
5
0
-5
-10
-15
-10
-5
0
5
10
15
40
20
0

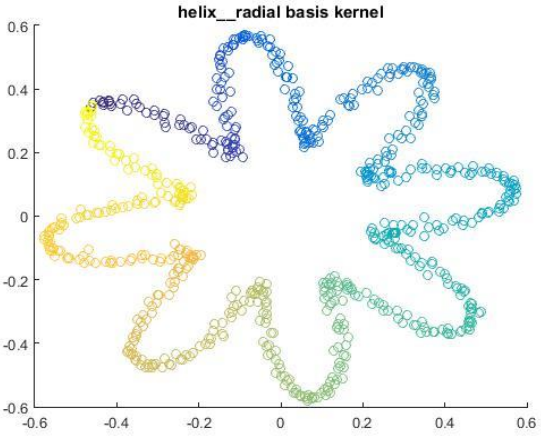
Part b

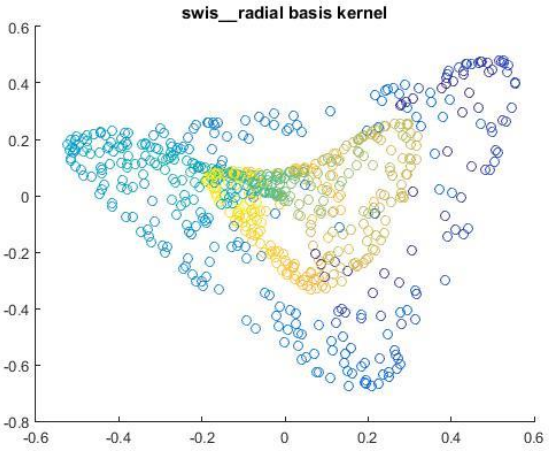

helix__linear kernel


swis__linear kernel


helix__polynomial kernel


swis__polynomial kernel
×10^4


helix__radial basis kernel


swis__radial basis kernel

Part c

As it is obvious in the pictures in part b, the embedding to lower dimensional act relatively the same for helix data (especially for linear and radial basis kernel). Also, compare to Swiss data, embedding acts better on helix data. As we can see, for Swiss data, embedding does not have a good performance and data are kind of mixed with all three kernels (especially polynomial one). Also for Swiss data different kernel acts differently. I think for both helix and Swiss data if want to rate the performance we can say

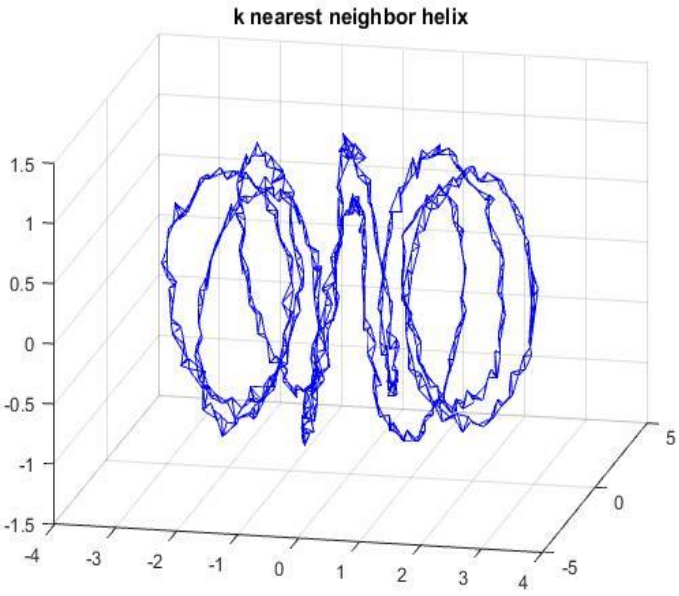
Radial basis kernel> linear kernel>polynomial kernel

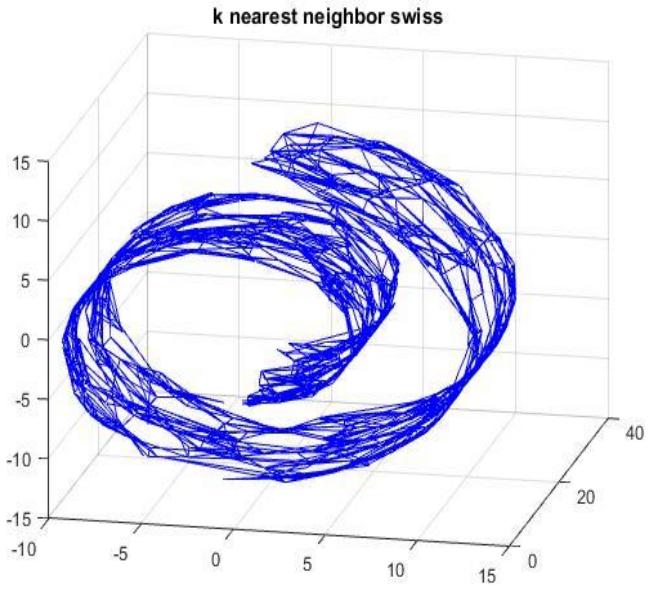
For saving paper I answer part e also here, the pictures of part d are in next page

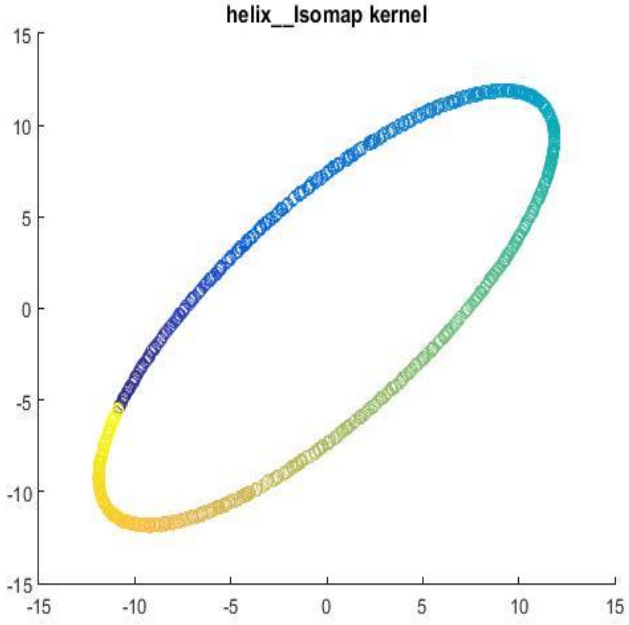
Part e

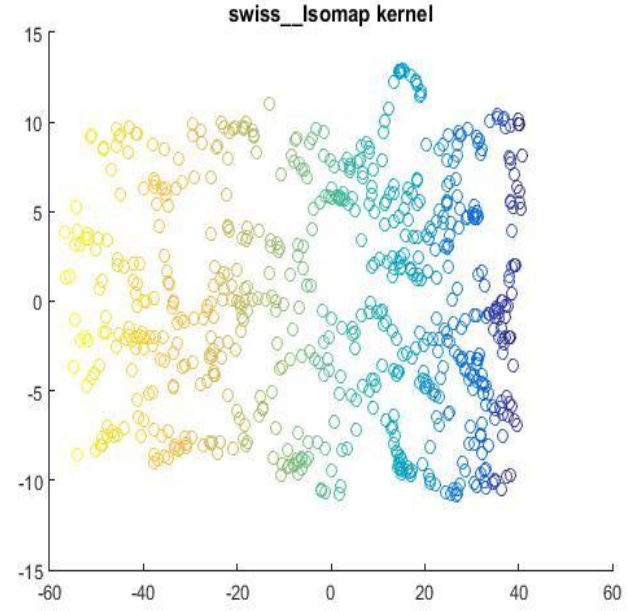
As we can see in picture of part d (next page), Isomap perfectly embedded both dataset to 2 dimensional set. So Isomap has a better performance compare to kernels that we use in part b. for example, as we can see for Swiss data purple and yellow data must be far apart from each other which is exactly the case in 2 dimensions.

Part d


k nearest neighbor helix


k nearest neighbor swiss


helix__lsomap kernel


swiss__lsomap kernel

Matlab code

athar_matlab5

```
%% Part a

load('helix.mat')
figure(1)
grid on
scatter3(X(1,:),X(2,:),X(3,:),[],tt)
view([12 20])
title('helix data')
clear X tt

load('swiss')

figure(2)
grid on
scatter3(X(1,:),X(2,:),X(3,:),[],tt)
view([12 20])
title('swiss data')
%% Part B
clear
clc
load('helix.mat')


K_linear=X'*X;
K_poly=(X'*X).^3;
 A=X';
for j=1:700

    Xm = bsxfun(@minus,A,A(j,:));
    W(:,j) = dot(Xm,Xm,2);

end
rng('default')
K_radial=exp(W./(-2*4^2));

H=(eye(700)-(1/700)*(ones(700,1)*ones(1,700)));

K_linear_G=H*K_linear*H;
K_poly_G=H*K_poly*H;

K_radial_G=H*K_radial*H;

[V1,D1] = eigs(K_linear_G);

[~,I] = sort(diag(D1),'descend');
V1=V1(:,I(1:2));
D1=diag(D1);
```

```
lamda1=(diag(D1(I(1:2)))).^0.5;

[V2,D2] = eigs(K_poly_G);
[~,I] = sort(diag(D2),'descend');
V2=V2(:,I(1:2));
D2=diag(D2);
lamda2=(diag(D2(I(1:2)))).^0.5;

[V3,D3] = eigs(K_radial_G);
[~,I] = sort(diag(D3),'descend');
V3=V3(:,I(1:2));
D3=diag(D3);
lamda3=(diag(D3(I(1:2)))).^0.5;

X1=lamda1*V1';
X2=lamda2*V2';
X3=lamda3*V3';

figure(3)
scatter(X1(1,:),X1(2,:),[],tt)
title('helix__linear kernel')
figure(4)
scatter(X2(1,:),X2(2,:),[],tt)
title('helix__polynomial kernel')
figure(5)
scatter(X3(1,:),X3(2,:),[],tt)
title('helix__radial basis kernel')

%%
clear
clc

load('swiss.mat')


K_linear=X'*X;
K_poly=(X'*X).^3;
A=X';
for j=1:700

    Xm = bsxfun(@minus,A,A(j,:));
    W(:,j) = dot(Xm,Xm,2);

end
rng('default')
K_radial=exp(W./(-2*4^2));

H=(eye(700)-(1/700)*(ones(700,1)*ones(1,700)));

K_linear_G=H*K_linear*H;
K_poly_G=H*K_poly*H;

K_radial_G=H*K_radial*H;
```

```
[V1,D1] = eigs(K_linear_G);

[~,I] = sort(diag(D1),'descend');
V1=V1(:,I(1:2));
D1=diag(D1);
lamda1=(diag(D1(I(1:2)))).^0.5;

[V2,D2] = eigs(K_poly_G);
[~,I] = sort(diag(D2),'descend');
V2=V2(:,I(1:2));
D2=diag(D2);
lamda2=(diag(D2(I(1:2)))).^0.5;

[V3,D3] = eigs(K_radial_G);
[~,I] = sort(diag(D3),'descend');
V3=V3(:,I(1:2));
D3=diag(D3);
lamda3=(diag(D3(I(1:2)))).^0.5;

X1=lamda1*V1';
X2=lamda2*V2';
X3=lamda3*V3';

figure(6)
scatter(X1(1,:),X1(2,:),[],tt)
title('swiss_linear kernel')
figure(7)
scatter(X2(1,:),X2(2,:),[],tt)
title('swiss_polynomial kernel')
figure(8)
scatter(X3(1,:),X3(2,:),[],tt)
title('swiss_radial basis kernel')

%% Part d
%i)
%for helix
clear
clc
load('helix.mat')
X_1=(pdist2(X',X'));
euc_X1=zeros(700,700);
for i=1:700
[S,I]=sort(X_1,2);
euc_X1(i,I(i,1:8))=X_1(i,I(i,1:8));

end

for i=1:700
    figure(9)
grid on
II=euc_X1(i,:)>0;
plot3(X(1,II),X(2,II),X(3,II),'b')
view([12 20])

    hold on
```

```
end
title('k nearest neighbor helix')

hold off




%ii)
 D=dijk(euc_X1,[1:700],[1:700]);
 D=D.^2;
H=(eye(700)-(1/700)*(ones(700,1)*ones(1,700)));
K_Isomap_G=(-0.5)*H*D*H;
rng('default')
[V1,D1] = eigs(K_Isomap_G);

[~,I] = sort(diag(D1),'descend');
V1=V1(:,I(1:2));
D1=diag(D1);
lamda1=(diag(D1(I(1:2)))).^0.5;
X1=lamda1*V1';
figure(11)
scatter(X1(1,:),X1(2,:),[],tt)
title('helix__Isomap kernel')



%for swiss
clear
clc
load('swiss.mat')
X_1=(pdist2(X',X'));
euc_X1=zeros(700,700);
for i=1:700
[S,I]=sort(X_1,2);
euc_X1(i,I(i,1:8))=X_1(i,I(i,1:8));

end

for i=1:700
    figure(10)
grid on
II=euc_X1(i,:)>0;
plot3(X(1,II),X(2,II),X(3,II),'b')
view([12 20])

    hold on
end
title('k nearest neighbor swiss')
hold off


% delta=D.^2;
```

```
%ii)
D=dijk(euc_X1,[1:700],[1:700]);
D=D.^2;
H=(eye(700)-(1/700)*(ones(700,1)*ones(1,700)));
K_Isomap_G=(-0.5)*H*D*H;
rng('default')
[V1,D1] = eigs(K_Isomap_G);

[~,I] = sort(diag(D1),'descend');
V1=V1(:,I(1:2));
D1=diag(D1);
lamda1=(diag(D1(I(1:2)))).^0.5;
X1=lamda1*V1';
figure(12)
scatter(X1(1,:),X1(2,:),[],tt)
title('swiss__Isomap kernel')
```