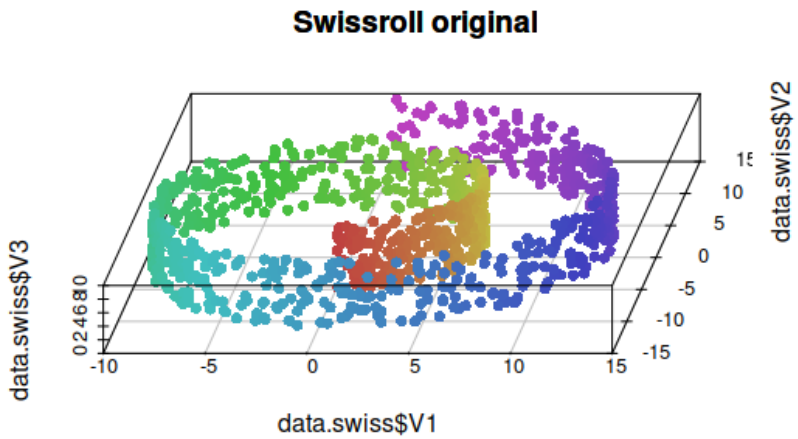
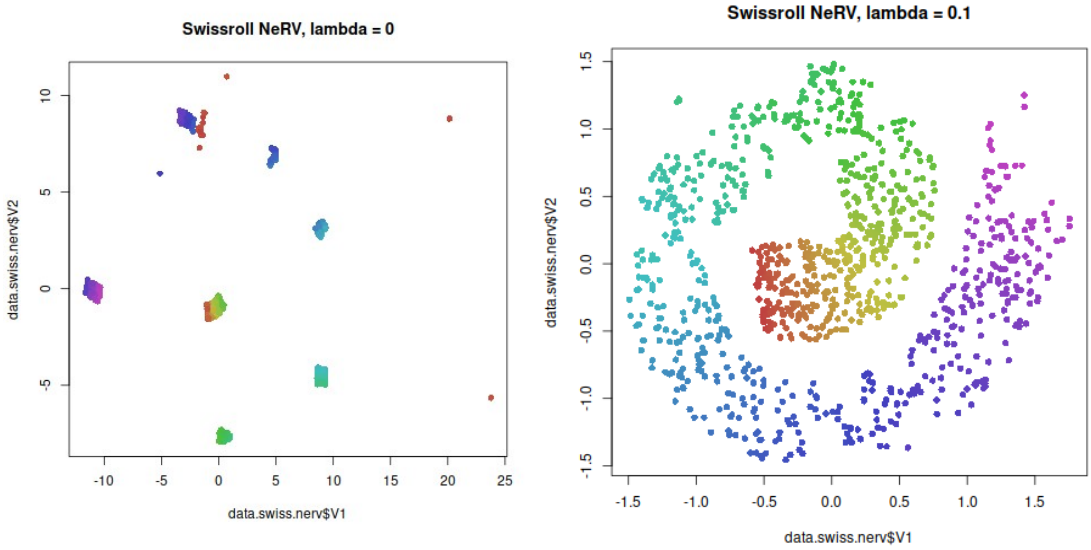
DATA.STAT.770

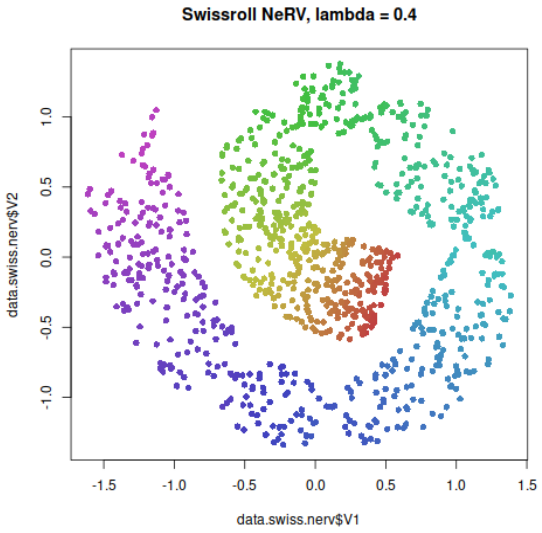
Exercise set 12

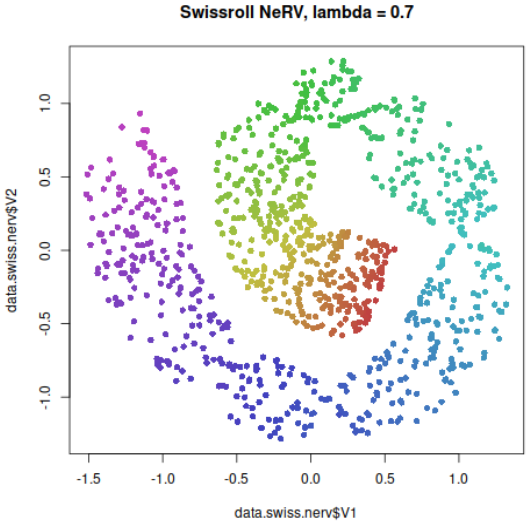
G2.

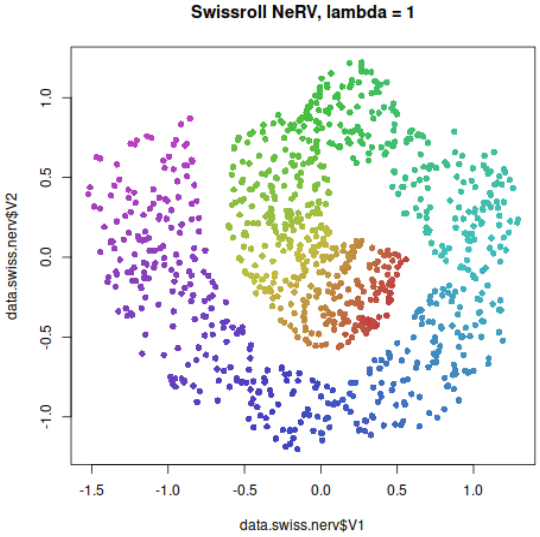
Swissroll dataset visualization:



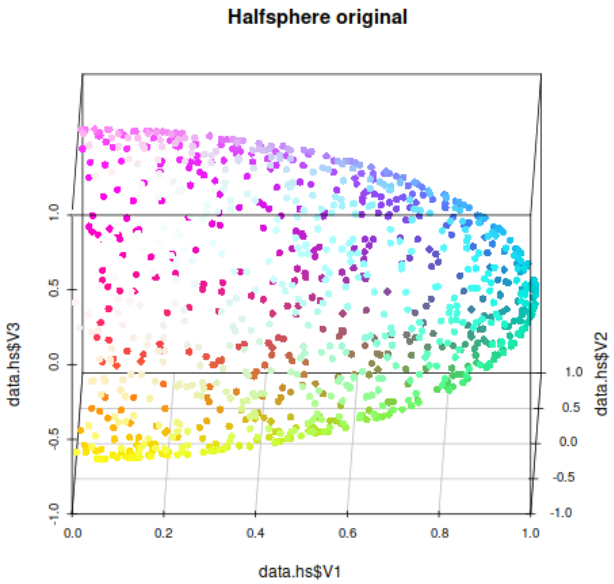


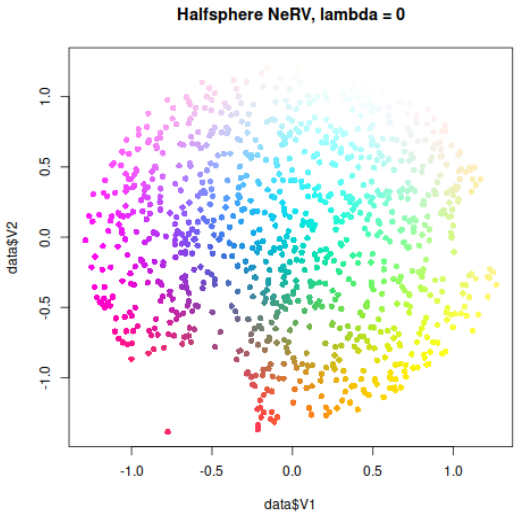


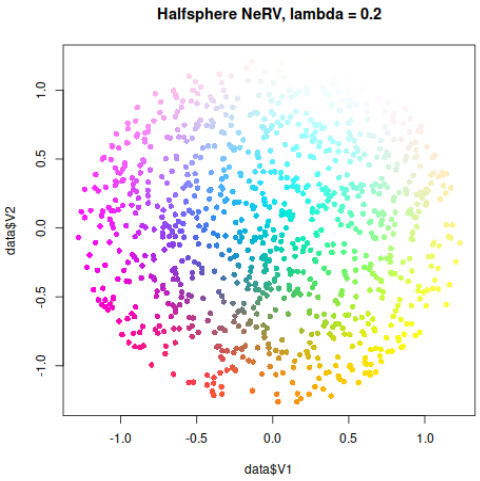


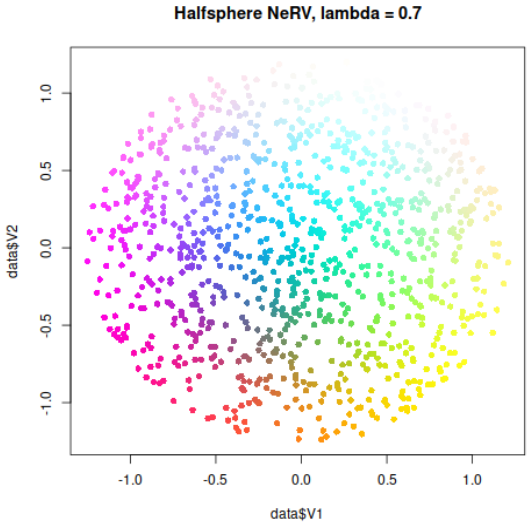


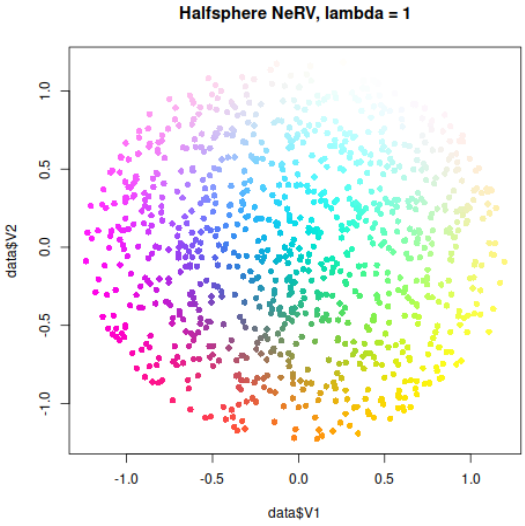
Halfsphere dataset:



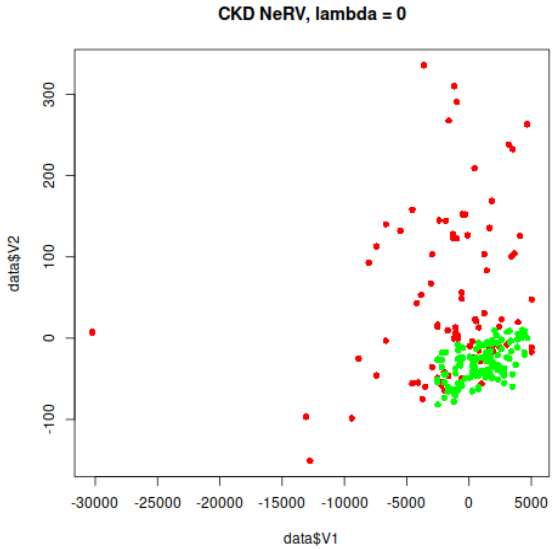

Halfsphere NeRV, lambda = 0
data$V2
data$V1
-1.0
-0.5
0.0
0.5
1.0

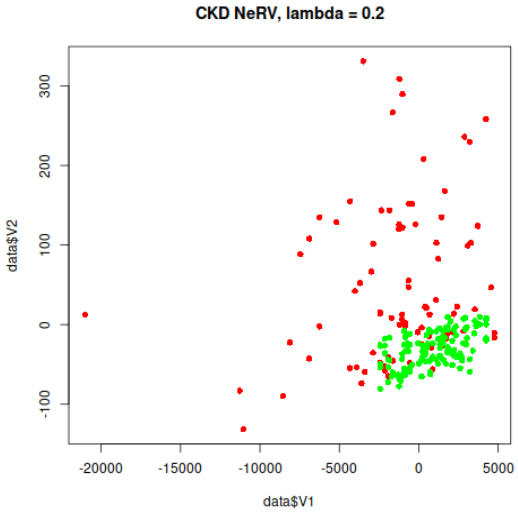

Halfsphere NeRV, lambda = 0.2
data$V2
data$V1
-1.0
-0.5
0.0
0.5
1.0

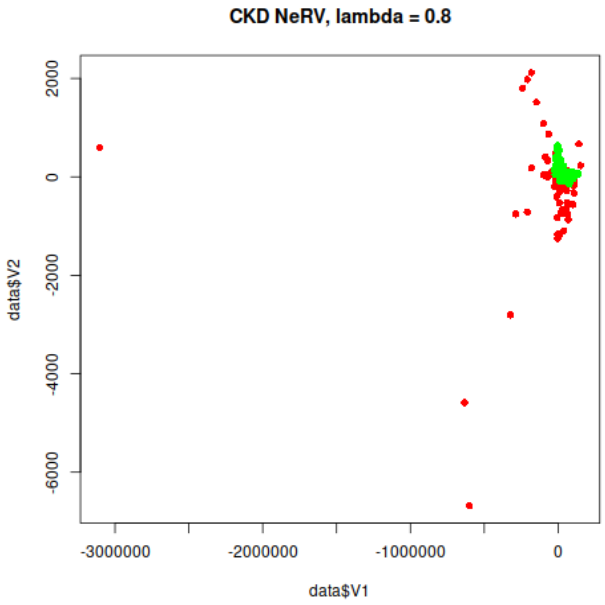

Halfsphere NeRV, lambda = 0.7
data$V2
data$V1
-1.0
-0.5
0.0
0.5
1.0

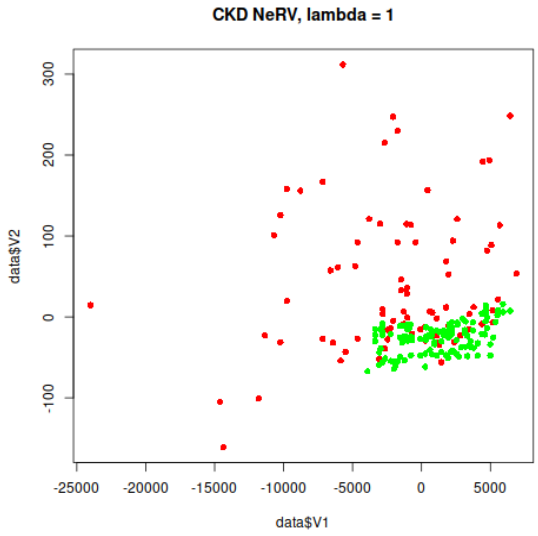

Halfsphere NeRV, lambda = 1
data$V2
data$V1
-1.0
-0.5
0.0
0.5
1.0

CKD dataset:


CKD NeRV, lambda = 0
data$V2
data$V1
-100
0
100
200
300
-30000
-25000
-20000
-15000
-10000
-5000
0
5000

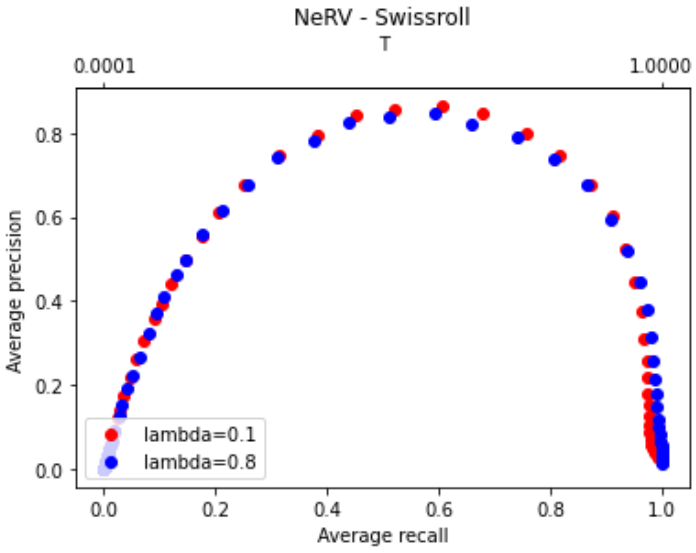

CKD NeRV, lambda = 0.2
data$V2
data$V1
-100
0
100
200
300
-20000
-15000
-10000
-5000
0
5000

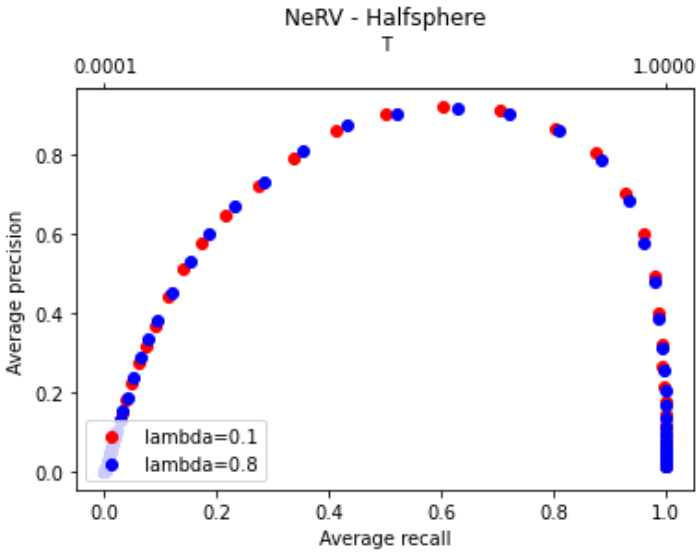


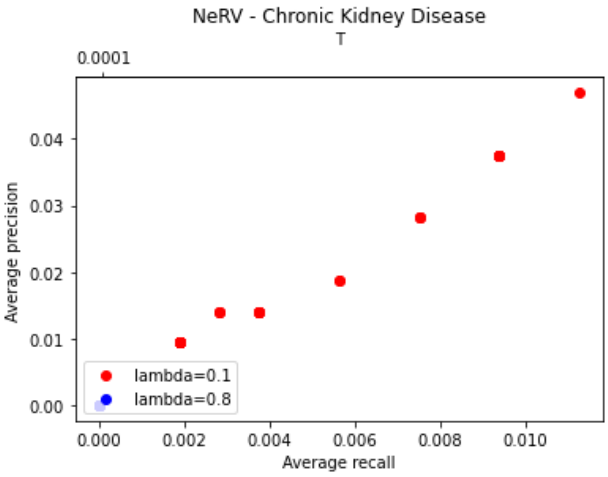


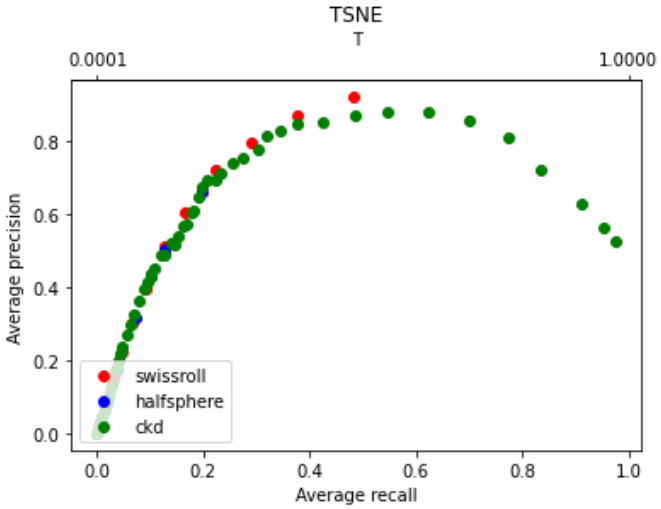
In most cases the small lambda parameter adjustment doesn't affect dramatically to the output. The most interesting patterns maybe happened in swissroll dataset. In the case of lambda=0 (full precision) the visualization merely produces individual clusters instead of unfolding.

Precision and recall curves. In cases where 0 neighbours were retrieved, I set precision to 0. T-values (neighbourhood sizes) are from [0.001, 1] and they are decreased iteratively by multiplying by 0.9.

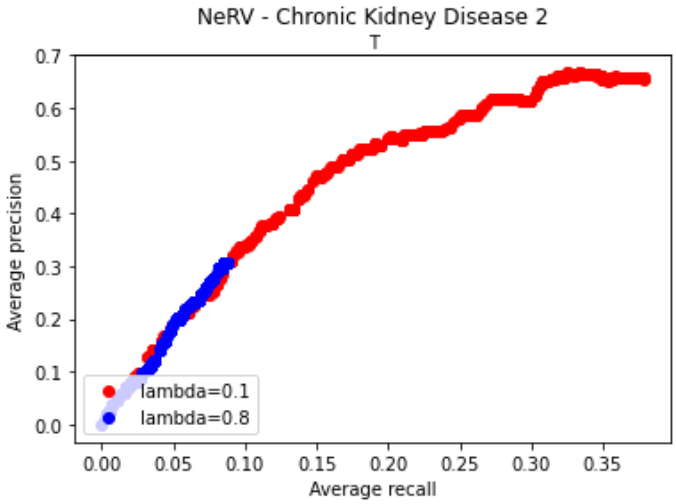






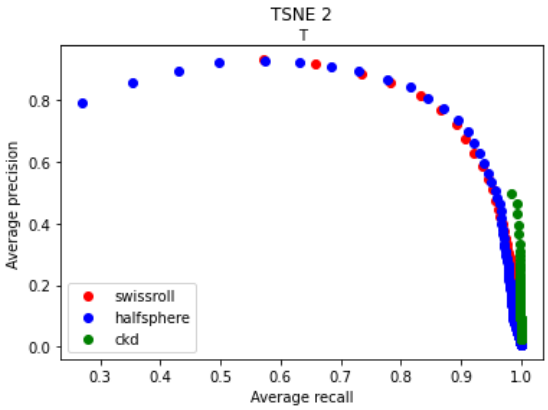


It seems that NeRV method performed pretty well in terms of precision and recall at swissroll and halfsphere datasets. However, originally 14-dimensional CKD dataset causes some difficulties. Overall it seems that projection produced by NeRV is rather sparse (as can be seen from point clouds). Especially when lambda=0.8, both precision and recall stayed at 0. I changed the neighbourhood size from [0.001,1] to [1.1, 50], with step size of 1, and obtained better results as seen below. I had some issues with top T-axis, but the left side means T=1.1 and right side T=50.



The performance is still poorer compared to swissroll and halfsphere, but better than previously.

Lastly, I used TSNE instead of SNE. Compared to the NeRV, this method performed better at CKD data but worse at swissroll and halfsphere (lot of false negatives compared to true positives). I did the same trick mentioned previously (changed T-scale) and obtained following results



The results were improved for swissroll and halfsphere, but CKD preformance in precision decreased remarkably. Based on these results we can deduce that TSNE and NeRV projections operate very differently: NeRV projections for swissroll and halfsphere are quite dense but TSNE projections for these datasets are sparse. Alternatively NeRV projection for CKD is sparse but corresponding TSNE projection is quite dense.

G3.

Q1:

By the strict definition of precision and recall, good precision denotes a property that that the close-by points in the projected dataset are also close in original dataset, and good recall denotes that close-by points in the original dataset are also close in the projected dataset. Based on these definitions, both precision and recall focuses to preserving neighbourhoods that are defined as closeness of samples where closeness is defined in terms of distances. That said it is clear that neighbourhoods and distances are closely related and when the general scale of distances is same in both original and reduced spaces, we can argue that precision and recall essentially measures same as distance preservation. However, when the general scales differ space to space the algorithms focusing on precision and recall or distance preservation might produce very different results. For example, say our objective emphasizes the preservation of original neighbourhoods in terms of precision and recall. Assume that in the original space, point x_i has closest samples (neighbourhood) $x_{i1}, ..., x_{in}$ and our dimensionality reduction algorithm produces corresponding projections y_i and $y_{i1}, ..., y_{in}$, where $x_{i\cdot} \rightarrow y_{i\cdot}$ and $y_{i1}, ..., y_{in}$ is the neighbourhood of y_i. Now we can see that the precision and recall for sample y_i are very good (as a matter of fact 1) because all original neighbourhood samples were preserved. However, it might be that $d(x_i, x_{i\cdot}) < d(y_i, y_{i\cdot})$ or $d(x_i, x_{i\cdot}) > d(y_i, y_{i\cdot})$, because essentially precision and recall measures only the performance in information retrieval (it is enough to have same samples close to the target point in respective spaces), but not in distance preservation. To summarize: precision and recall doesn't essentially mean/measure the same thing as distance preservation, sometimes they might produce similar results, but in general no.

Q2:

Curvilinear Component Analysis is a dimensionality reduction method that focuses on obtaining the property $d(y_i, y_j)) = \text{ small }_y \Rightarrow d(x_i, x_j)) = \text{ small }_x$ where y's denotes points at reduced space and $small_y$ denotes the small distances in reduced space and $small_x$ in original space. Because CCA directly compares the distances regardless respective space, in this particular case $small_y = small_x$. Because the scales of 'smallness' are same, based on discussion on Q1 it is obvious that CCA performs well in terms of precision. Hence there is no conflicts with the above discussion and CCA projection.