Homework 3: SVD Imputation & Kernel PCA

Theresa Wakonig
Data Mining II

April 21, 2021

1 Exercise 1: Data Imputation via SVD

1.a

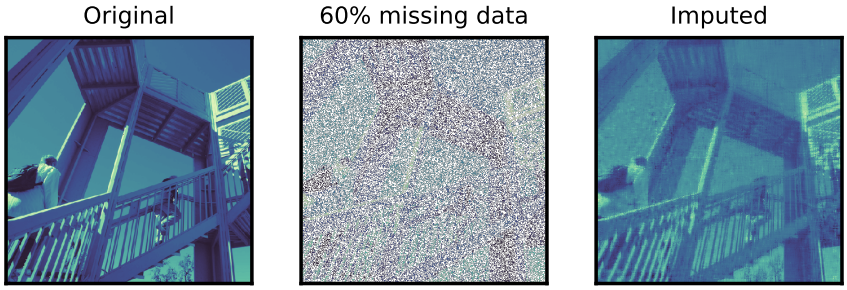

Figure 1: Results of SVD imputation using `svd_imputation_optimised()`.

1.b

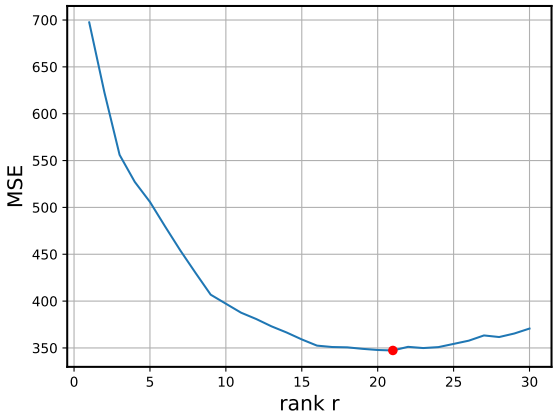

Figure 2: Tested ranks r and the corresponding mean squared errors.

The minimum, marked with a red dot, depicts the optimal rank to choose for imputing the given image via SVD.
Optimal rank = 21
Corresponding minimum mean-squared error: 347.32

1.c

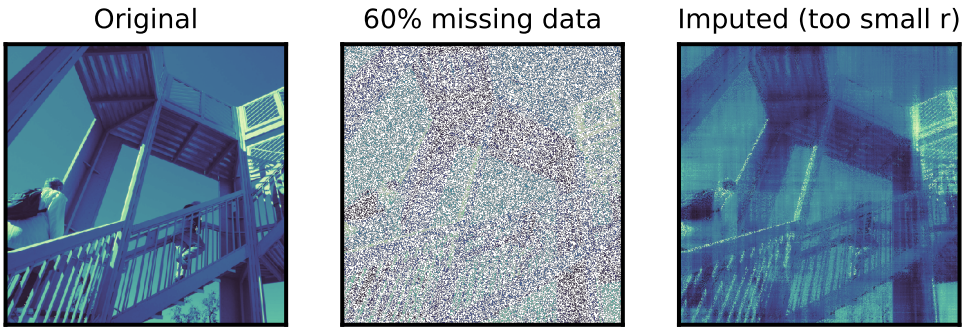

Figure 3: SVD imputation with rank $r = 5$.

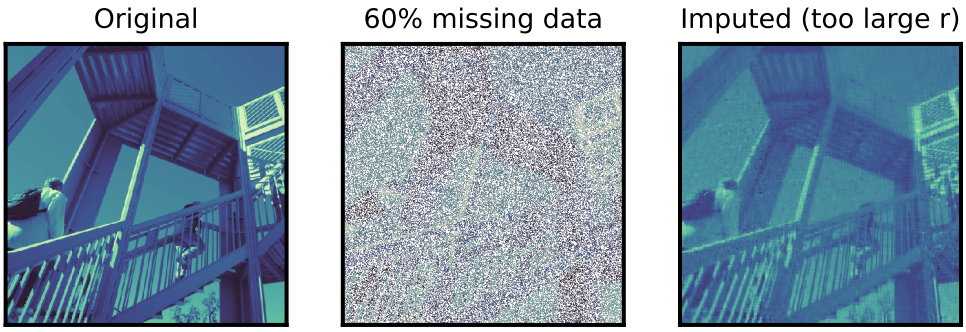

Figure 4: SVD imputation with rank $r = 29$.

Choosing too large or too small values for the rank will deteriorate the quality of the imputed image, as can be seen in the images above. Truncating the rank-1 sum expansion too early (small rank approximation of our matrix) will result in losing information. The image appears less detailed. Using rank values that are too big will however lead to more noise being fed into the imputation process, causing the MSE to rise. Moreover, choosing a large rank r results in longer computation times (more iterations until convergence).

2 Exercise 2: Kernel PCA

2.a

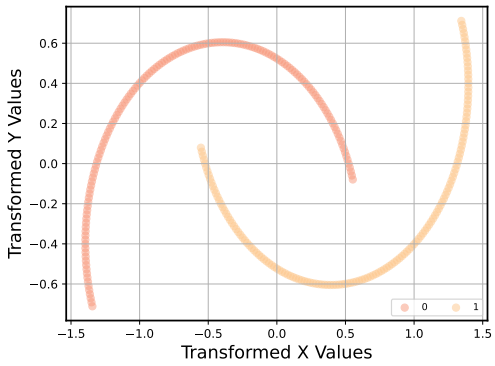

Figure 5: PCA on the half-moon data using the `computePCA()` function.

Applying the linear form of PCA to this non-linear dataset does, as expected, not separate the two classes. The first two principal components do not allow linear separation of the two classes, which is clearly visible when projecting the points on either of the two axes. There is more variation of the data in the x-coordinate which represents the first PC.

2.b

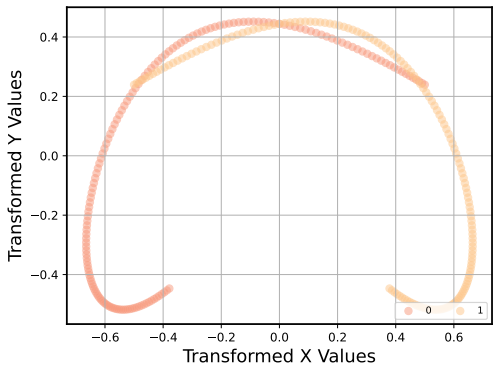

Figure 6: PCA on the half-moon data using the function `RBFKernelPCA()` and $\gamma = 1$.

The plot depicts the transformed data when applying the Kernel PCA with RBF Kernel and using the value 1 for the hyperparameter γ. The two classes are not separable.

2.c

The Kernel PCA allows us to perform non-linear dimensionality reduction which is more suitable for this dataset than the linear form of PCA used in 2a). As expected, superior performance than in subtask a) can be observed (depending on γ). The plots depict a trend: a higher value of γ yields a more clear separation

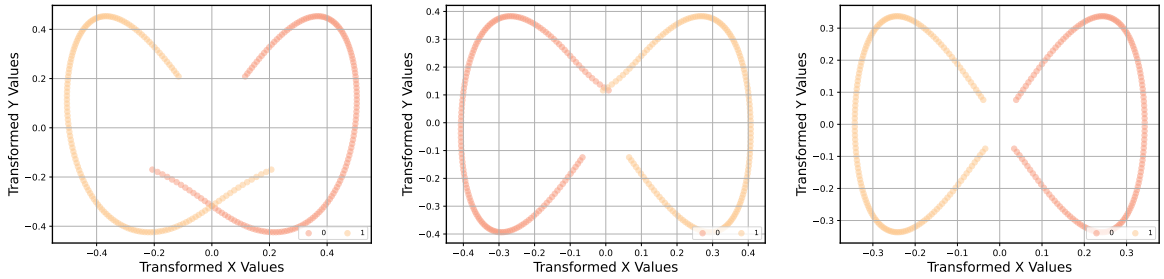

Figure 7: PCA on the half-moon data using the function `RBFKernelPCA()` and $\gamma = 5$, $\gamma = 10$, $\gamma = 20$ from left to right.

of the data. For $\gamma = 20$ we can see that the data points have been transformed in a way that the classes are nicely linearly separable and one could now use linear classifiers on the transformed data. I believe that γ can be tuned just like other hyperparameters using cross validation (as introduced in DM I) on the dataset.