Kernel principal component analysis (Kernel PCA)

Kernel principal component analysis (PCA) is an extension of principal component analysis (PCA) using techniques of kernel methods. Kernel principal component analysis maps the data to a higher dimension using kernels with their corresponding parameter.

- There are different choices of kernels: linear, poly, rbf, sigmoid, cosine
- Unlike PCA, Kernel PCA may not give perfect reconstruction
- There are many more free parameters: gamma, alpha

Syntax

An example of a *KernelPCA o*bject is given here:

kernel_pca = KernelPCA(kernel="rbf", gamma=10, fit_inverse_transform=True, alpha=0.1)

We can fit the model and make a prediction:

kernel_pca.fit(X_train)

score_kernel_pca = kernel_pca.transform(X_test)

Multi-Dimensional Scaling

Multi-Dimensional Scaling (MDS) is a family of algorithms, one version of which is Principal Component Analysis (PCA).

Like PCA, MDS can be used for dimensionality reduction; MDS can also be used to map complex differences into visual space. The difference is that MDS preserves the distance between data points. There are many types of distances called dissimilarity metrics.

Metric MDS

Metric MDS represents points in an embedding. It determines the embeddings by minimizing a distance metric.

Non-Metric MDS

In Non-Metric MDS, we apply a function f(.) to the distance metric before minimizing the distance metric.

We can create the object this way:

```
    embedding =  MDS(dissimilarity='precomputed', random_state=0, n_components=2,
metric=False)
```