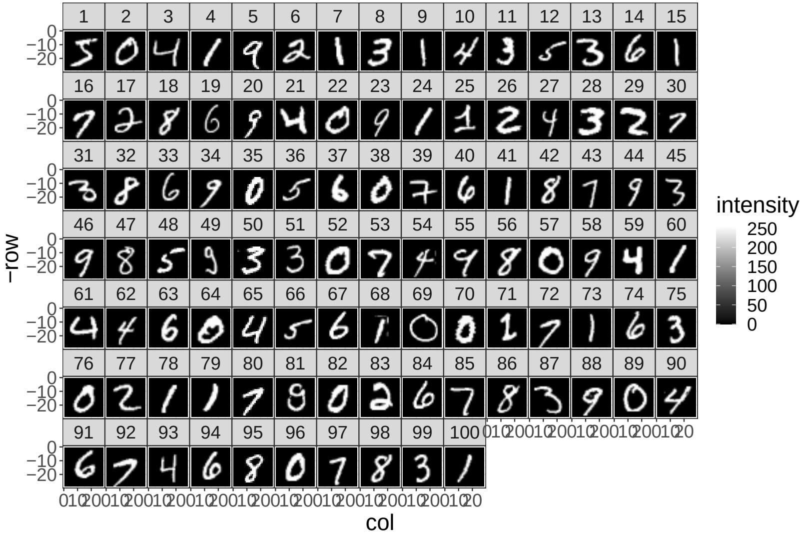
Other dimensionality reduction algorithms

Toby Dylan Hocking

Motivation: MNIST digits data



Set of digits is represented as a matrix

- Each digit image in MNIST data set is a matrix of 28×28 pixel intensity values, $x_i \in \{0, \ldots, 255\}^{784}$.
- Each of the images is a row in the data matrix.
- Each of the columns is a pixel.
- All images on last slide represented by a data matrix with $n = 100$ rows/images and $p = 784$ columns/pixels.

Background/motivation: non-linear dimensionality reduction

- High dimensional data are difficult to visualize.
- For example each observation/example in the MNIST data is of dimension 28 x 28 = 784 pixels.
- We would like to map each observation into a lower-dimensional space for visualization / understanding patterns in the data.
- Principal Components Analysis (PCA) is a linear dimensionality reduction method, which is computed using the Singular Value Decomposition (SVD).
- Auto-encoders are non-linear, which means they can be more accurate than PCA, in terms of reconstruction error.
- There are other non-linear dimensionality reduction methods.

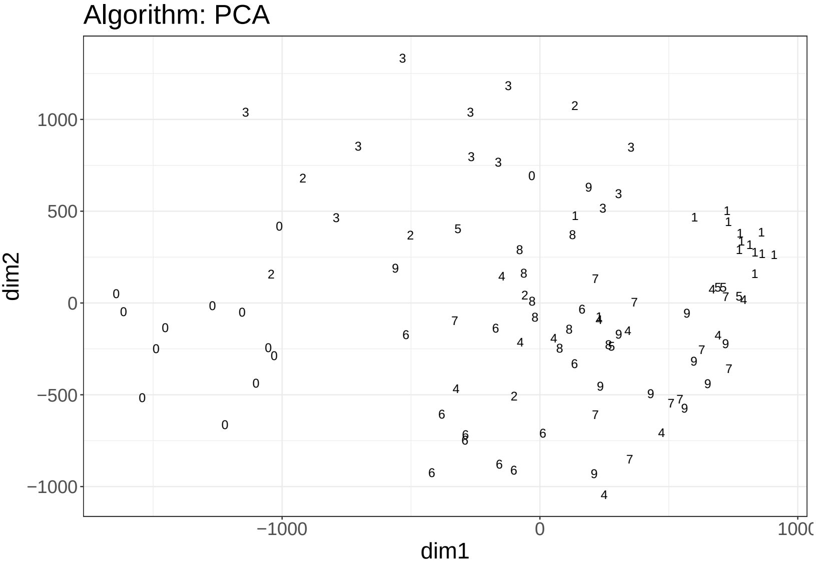
List of methods implemented in dimRed R package

- LLE: Local Linear Embedding.
- MDS: Multi-Dimensional Scaling.
- tSNE: t-distributed Stochastic Neighbor Embedding.

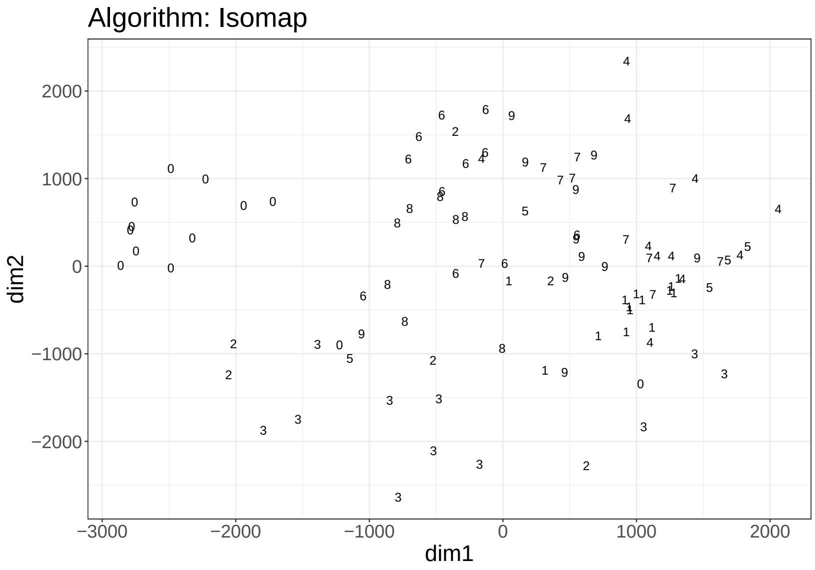
```
dimRed::dimRedMethodList()
```

```
##  [1] "AutoEncoder"            "DiffusionMaps"
##  [3] "DRR"                    "FastICA"
##  [5] "KamadaKawai"            "DrL"
##  [7] "FruchtermanReingold"    "HLLE"
##  [9] "Isomap"                 "kPCA"
## [11] "PCA_L1"                 "LaplacianEigenmaps"
## [13] "LLE"                    "MDS"
## [15] "nMDS"                   "NNMF"
## [17] "PCA"                    "tSNE"
## [19] "UMAP"
```

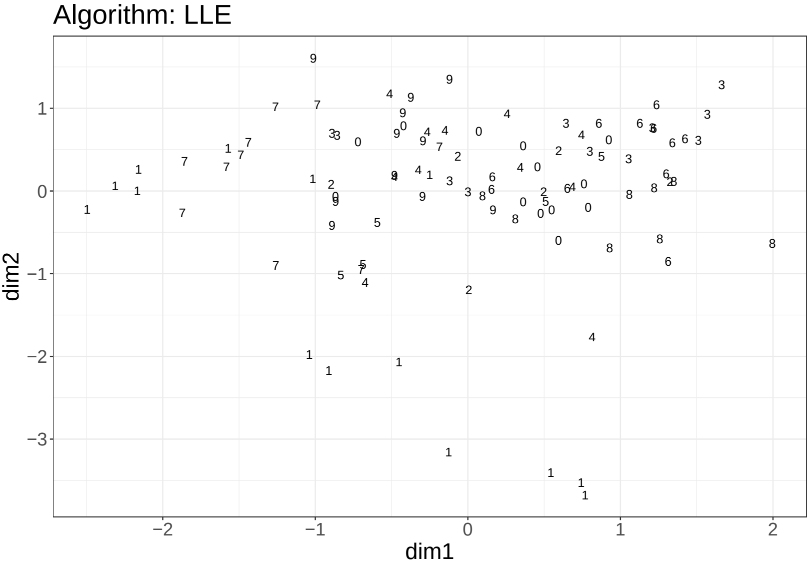
Comparison of different methods on MNIST data



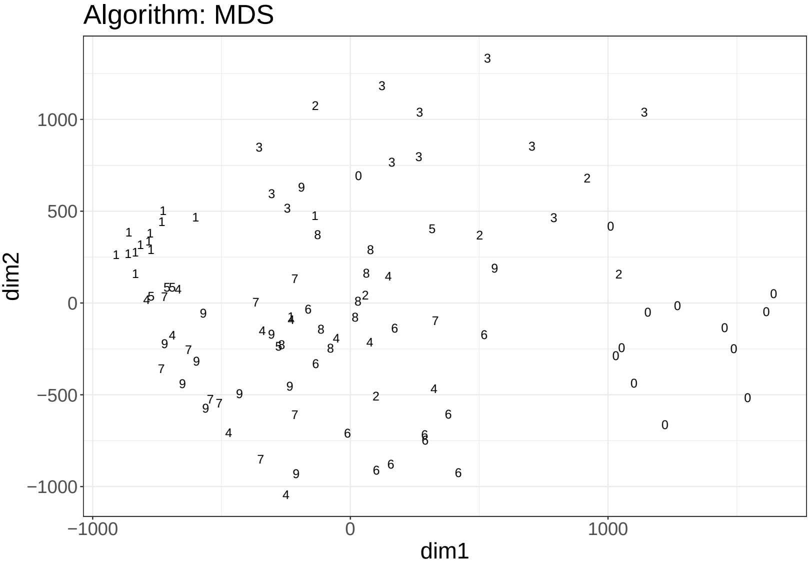
Comparison of different methods on MNIST data



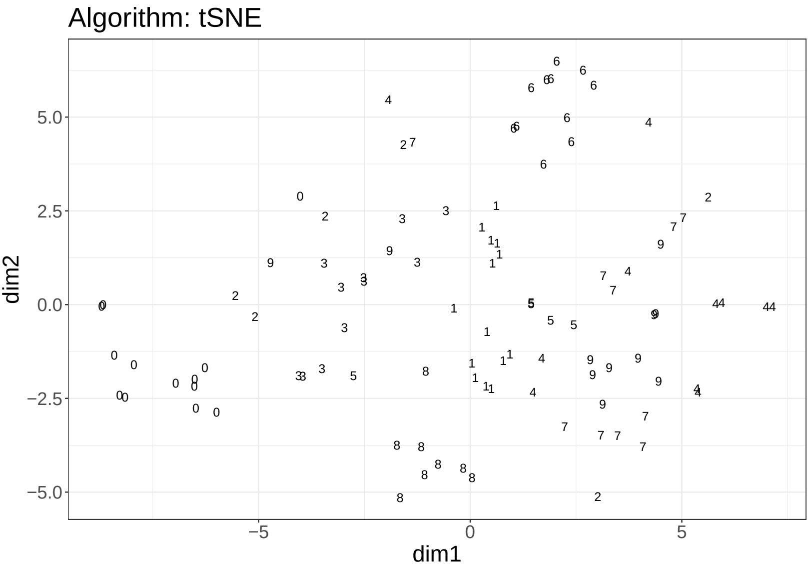
Comparison of different methods on MNIST data



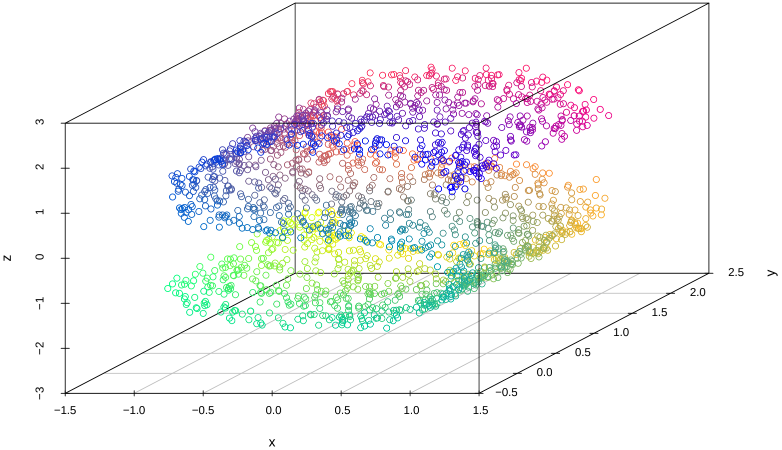
Comparison of different methods on MNIST data



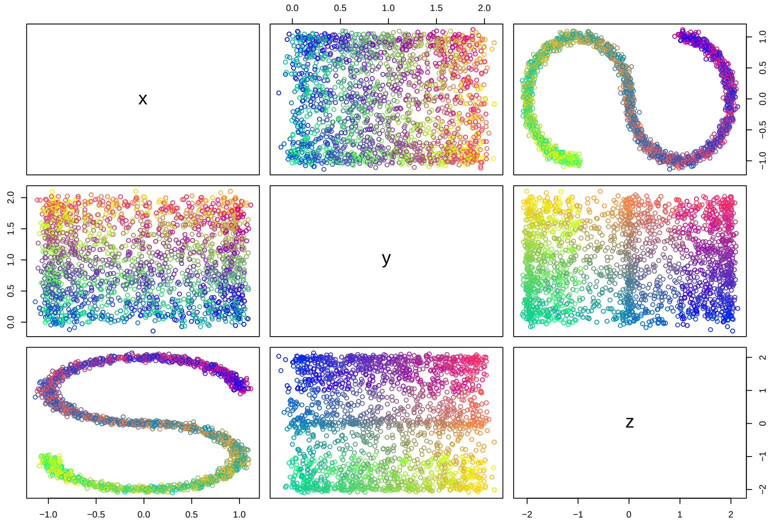
Comparison of different methods on MNIST data



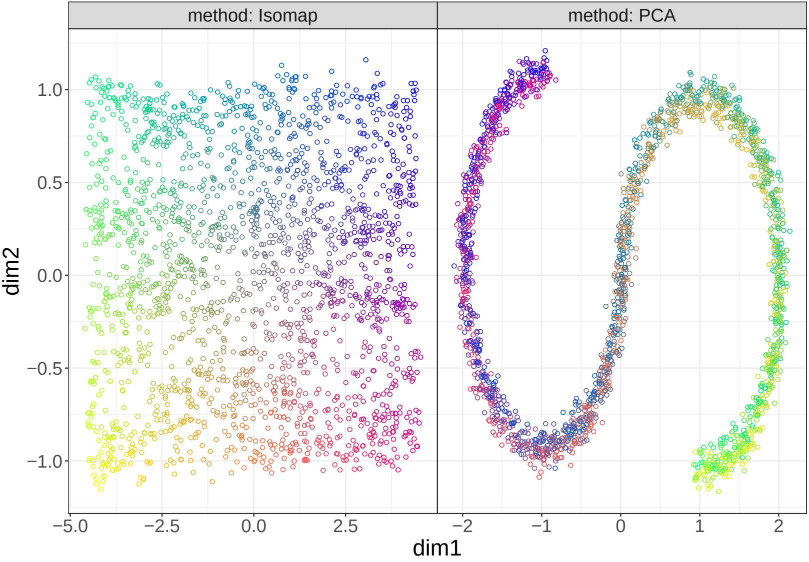
A toy data set in 3d with a clear 2d subspace



Scatter plot matrix view of same data



Two methods recover different low dimensional embeddings



How to evaluate/compare dimensionality reduction methods?

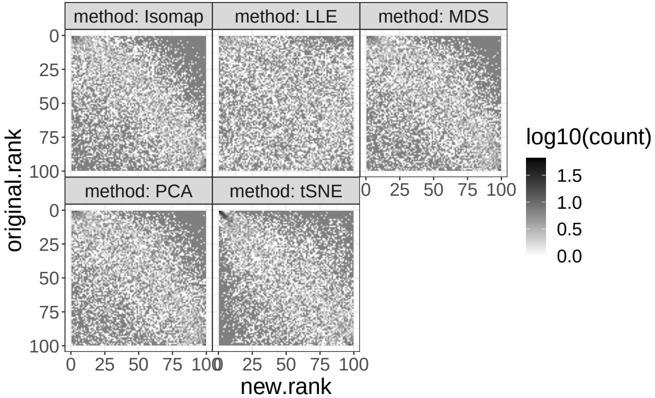
- Mean squared reconstruction error is what we used for PCA and auto-encoders.
- Whereas PCA and auto-encoders support decoding (function which inputs low dimensional values and returns high dimensional values in original space), many methods do not (LLE, tSNE, MDS).
- There are various other quality measures which only require the low-dimensional mapping (no need for decoder).

```
dimRed::dimRedQualityList()
```

```
## [1] "Q_local"                "Q_global"
## [3] "mean_R_NX"              "AUC_lnK_R_NX"
## [5] "total_correlation"      "cophenetic_correlation"
## [7] "distance_correlation"   "reconstruction_rmse"
```

Co-ranking matrix

- Many quality scores are based on the co-ranking matrix, which is based on computing rank distance matrices for both the high and low dimensional data, then q_{ij} is how many points of distance rank j became rank i.
- Diagonal co-ranking matrix is perfect, non-zeros in lower/upper triangle indicate points too close/far.



Comparing quality measures for MNIST

- These three quality scores are all based on co-ranking matrix (larger is better).
- tSNE method is best for these data.
- Default hyper-parameters were used for all methods.
- Better results for each algorithm could be obtained by choosing better hyper-parameters.

```
##         quality
## method    Q_local   Q_global AUC_lnK_R_NX
##   LLE    0.1691980 0.05309992    0.1415665
##   MDS    0.2494024 0.15783048    0.2600038
##   Isomap 0.2409395 0.17826925    0.2895633
##   tSNE   0.6498990 0.23398782    0.5608875
##   PCA    0.2494024 0.15783048    0.2600038
```

Possible exam questions

- TODO