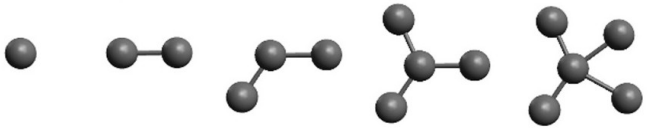
Dimensionality Reduction

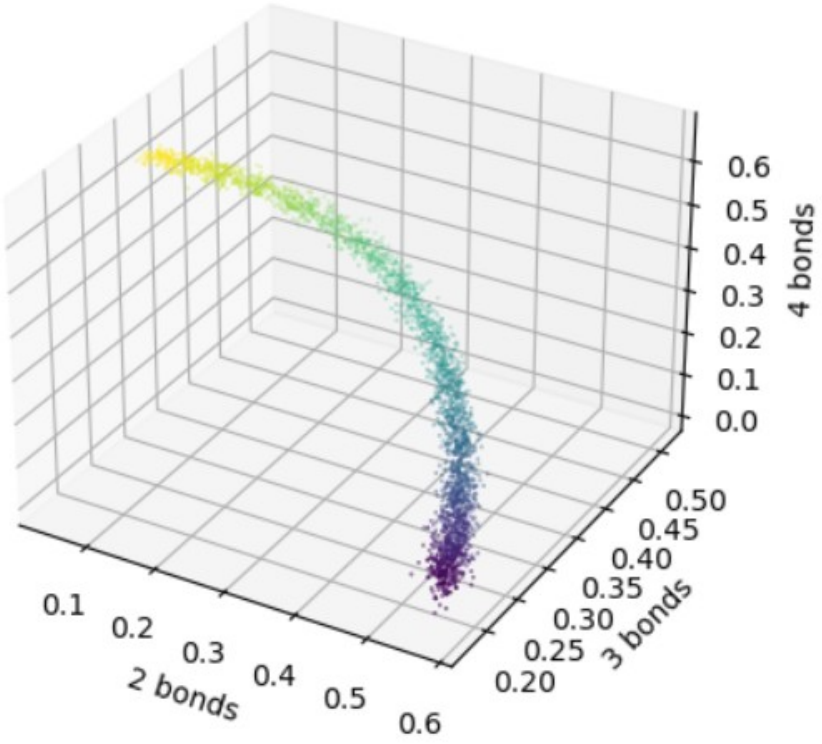
Vlad Gladkikh

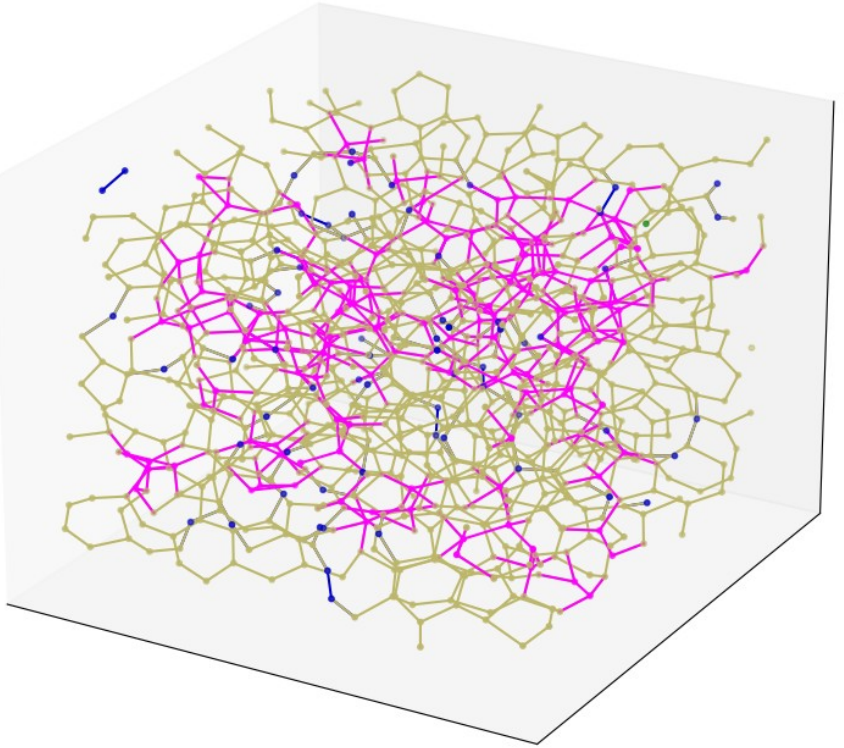
IBS CMCM

Data may be represented as points in a high-dimensional space
but span only a low-dimensional manifold.

K. Takahashi and Y. Tanaka. Unveiling descriptors for predicting the bulk modulus of amorphous carbon. Phys. Rev. B 95, 054110 (2017)







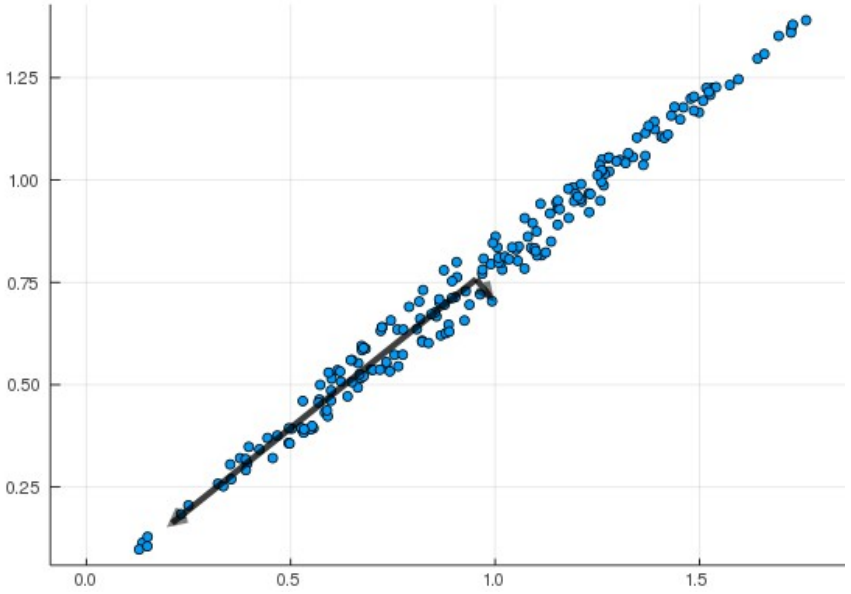
https://datascience.stackexchange.com/questions/89421/should-i-remove-a-feature-that-is-a-non-linear-function-of-another-feature-if-i

Dimensionality reduction task – reducing the number of features of a dataset while maintaining the essential relationships between the data points.

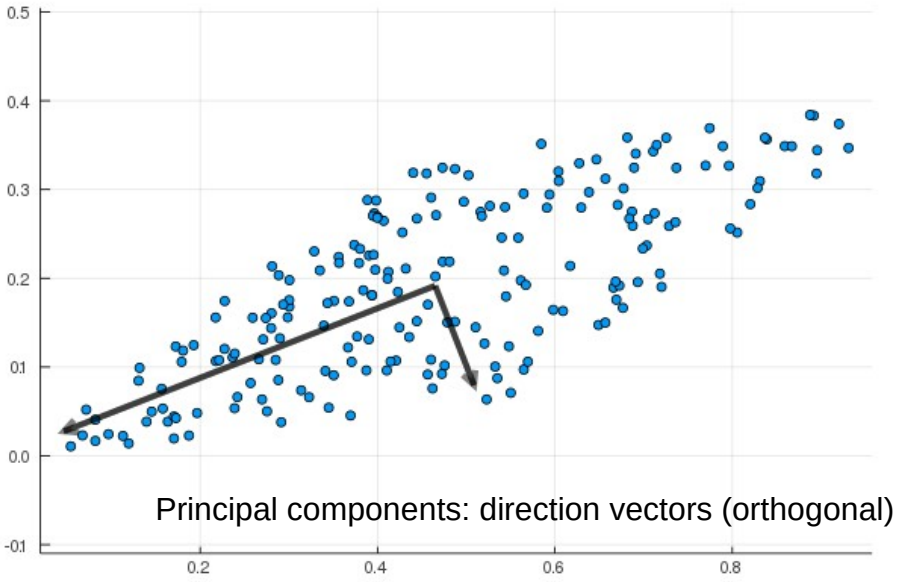
Principal Component Analysis (PCA)

Karl Pearson, 1901

PCA computes the orthogonal transform that decorrelates the variables and keeps the ones with the largest variance.



std ratio: 17.345



std ratio: 3.735

PCA for dimensionality reduction: zeroing out some principal components.

Dimensionality reduction loses information.

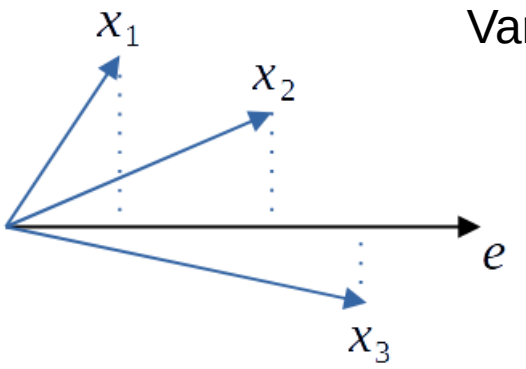
https://www.youtube.com/playlist?list=PLBv09BD7ez_5_yapAg86Od6JeeypkS4YM

https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

```
using MultivariateStats, Statistics, Plots
gr(fmt=:png);

n_data = 200
X1 = rand(2,2)
X2 = rand(2,n_data)
X = (X1 * X2)

M = fit(PCA, X; maxoutdim = 2, pratio = 1)

M_p = projection(M)
M_var = reshape(principalvars(M), (1,2))
M_std = sqrt.(M_var)
M_m = reshape(mean(M), (2,1))

println("projection(M)", size(M_p))
show(stdout, "text/plain", M_p); println('\n');

println("Orthogonal? ", sum(M_p[:,1] .* M_p[:,2]), '\n')
println("Norms: ", round( sum(M_p[:,1] .* M_p[:,1]) , digits=3),
              " ", round(sum(M_p[:,2] .* M_p[:,2]) , digits=3), '\n')

println("std(M) ", size(M_std))
show(stdout, "text/plain", M_std); println('\n');

println("Ratio: ", round( M_std[1] / M_std[2] , digits=3) ,'\n')

v = Array{Float64}(undef, (2,2+1))
v[:,1] = M_m
for i ∈ 1:size(M_p,2)
    v[:,i+1] = M_m .+ 2 * M_std[i] .* M_p[:,i]
end

scatter(X[1,:], X[2,:], legend = false, aspect_ratio=:equal)
plot!(v[1,[1,2]], v[2,[1,2]], line = (:arrow, 0.5, 4, :black))
plot!(v[1,[1,3]], v[2,[1,3]], line = (:arrow, 0.5, 4, :black))
```

Direction of greatest variability



Variance of projections:

$$\frac{1}{n-1}\sum_{i=1}^{n}\left(\sum_{j=1}^{p} x_{ij}\,e_j\right)^2$$

The sample mean of each feature has been shifted to zero

Maximize variance s.t. $\|e\|=1$

$$V=\frac{1}{n-1}\sum_{i=1}^{n}\left(\sum_{j=1}^{p} x_{ij}\,e_j\right)^2-\lambda\left(\left(\sum_{j=1}^{p} e_j^2\right)-1\right)$$

$$\frac{\partial V}{\partial e_a}=\frac{2}{n-1}\sum_{i=1}^{n}\left(\sum_{j=1}^{p} x_{ij}\,e_j\right)x_{ia}-2\lambda e_a=0$$

$$\sum_{j=1}^{p}\left(\frac{1}{n-1}\sum_{i=1}^{n} x_{ia}\,x_{ij}\right)e_j=\lambda\,e_a$$

$$\frac{1}{n-1}X^T X\,e=\lambda\,e$$

Yu-Shen Liu, Karthik Ramani,. Robust principal axes determination for point-based shapes using least median of squares.
https://doi.org/10.1016/j.cad.2008.10.012

https://math.stackexchange.com/questions/3869/what-is-the-intuitive-relationship-between-svd-and-pca

https://stats.stackexchange.com/questions/134282/relationship-between-svd-and-pca-how-to-use-svd-to-perform-pca

X: $n \times p$ *n* samples, *p* features

the sample mean of each column has been shifted to zero

Principal components: eigenvectors of $X^T X$ – covariance matrix

Covariance between two features shows if they change together or in the opposite direction.

Singular value decomposition of X: $X = U \Sigma W^T$

U: $n \times n$ orthogonal, its columns are called the left singular vectors of X

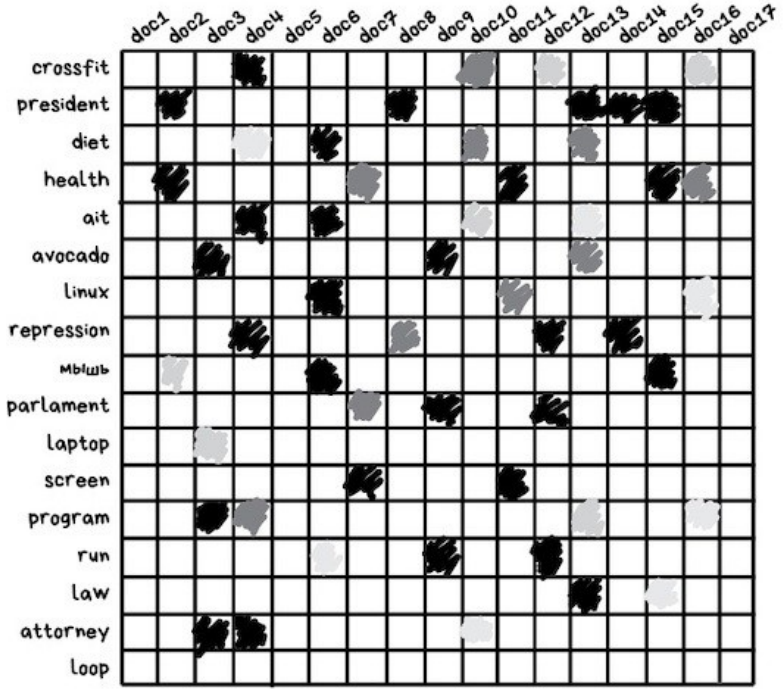
Σ: $n \times p$ rectangular diagonal matrix of positive numbers called the singular values of X

W: $p \times p$ orthogonal, its columns are called the right singular vectors of X

```
using LinearAlgebra

Xm = X .- mean(X, dims=2)  # X (p, n_data)
ei = eigen(Xm * Xm')
ei.values
ei.vectors

s = svd(Xm)
s.U # pc
s.S
s.Vt

# std
s.S/sqrt(n_data-1)
```

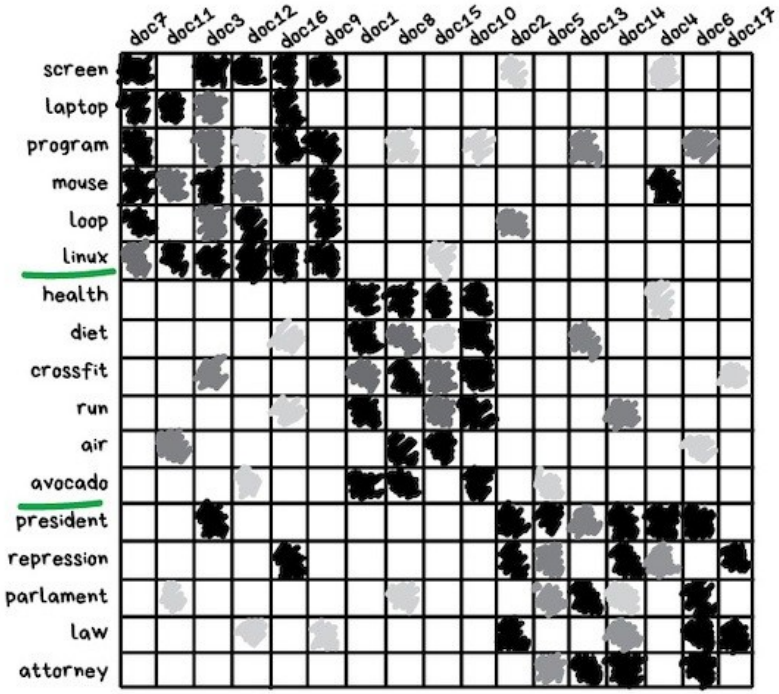
https://intoli.com/blog/pca-and-svd/

SEPARATE DOCUMENTS BY TOPIC



SVD

2. Transform



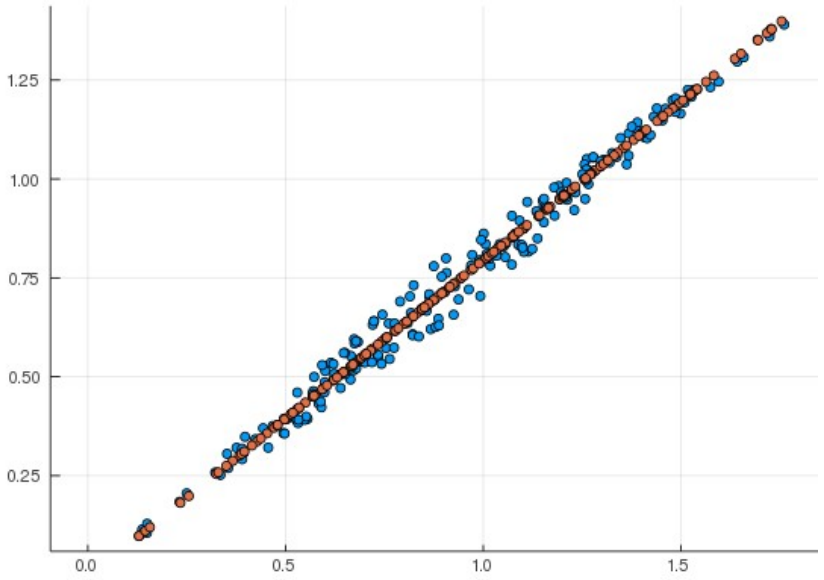
1. Build a matrix of how often each word can be found in each document
(black - more often)

3. Get visual topic clusters.
Even if the words haven't met together

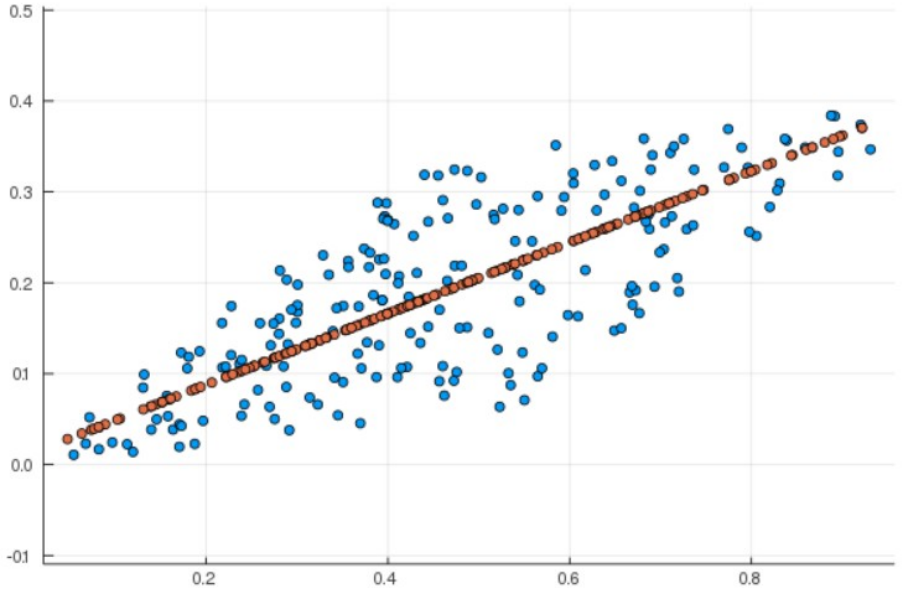
LATENT SEMANTIC ANALYSIS (LSA)

https://vas3k.com/blog/machine_learning/

The first principal component is a direction that maximizes the variance of the projected data.



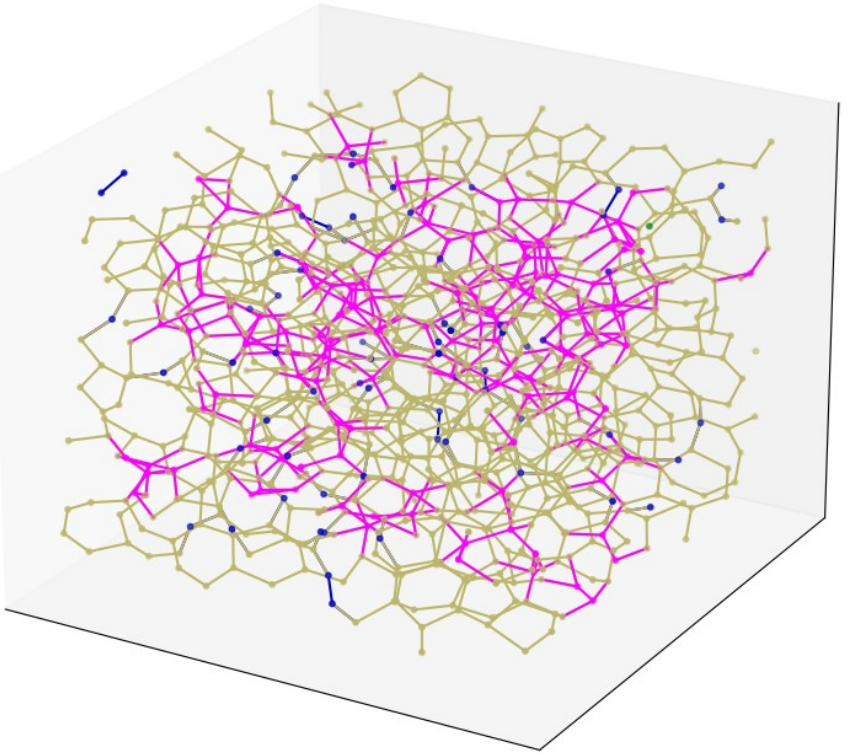
std ratio: 17.345

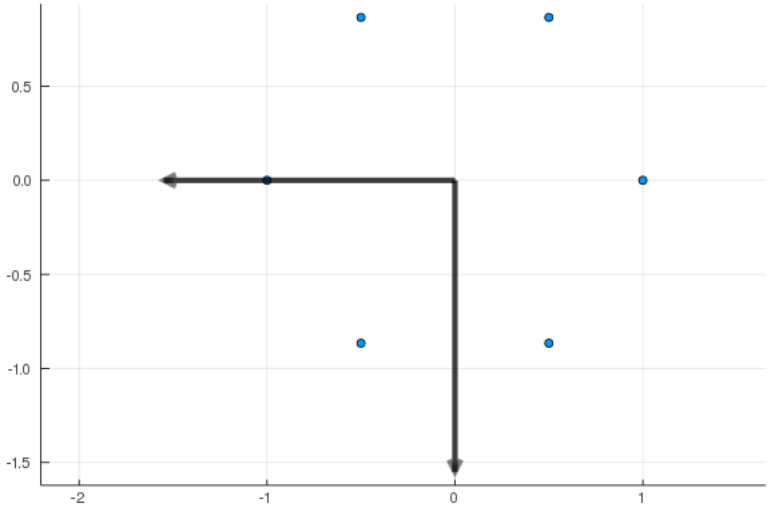


std ratio: 3.735

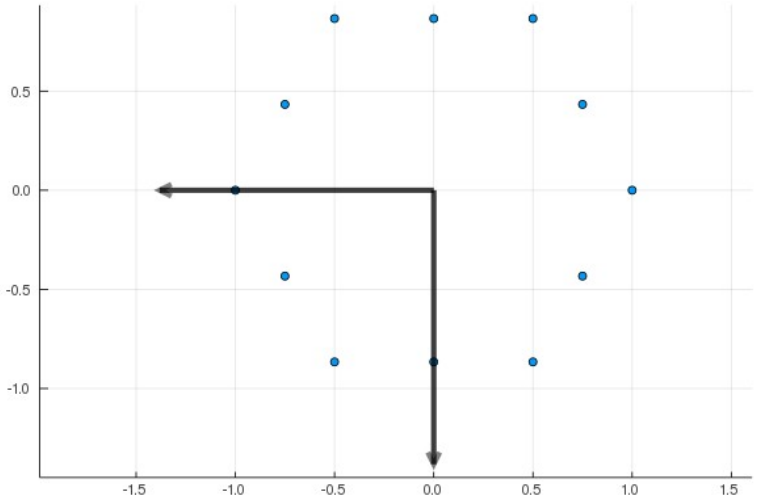
```
M = fit(PCA, X; maxoutdim = 1, pratio = 1)
Y = transform(M, X)
Xr = reconstruct(M, Y)

scatter(X[1,:], X[2,:], legend = false, aspect_ratio=:equal)
scatter!(Xr[1,:], Xr[2,:], legend = false, aspect_ratio=:equal)
```

The i th principal component can be taken as a direction orthogonal to the first $i - 1$ principal components that maximizes the variance of the projected data.





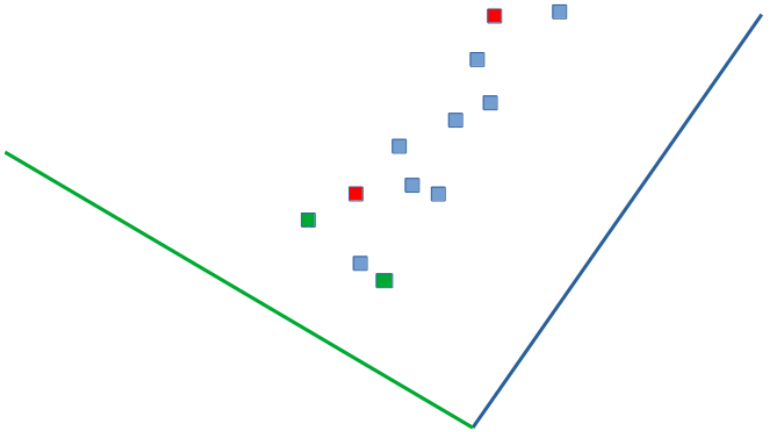
$$X^T X = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$$



$$X^T X = \begin{pmatrix} 5.25 & 0 \\ 0 & 5.25 \end{pmatrix}$$

https://ai.stackexchange.com/questions/27004/why-does-pca-of-the-vertices-of-a-hexagon-result-in-principal-components-of-equa

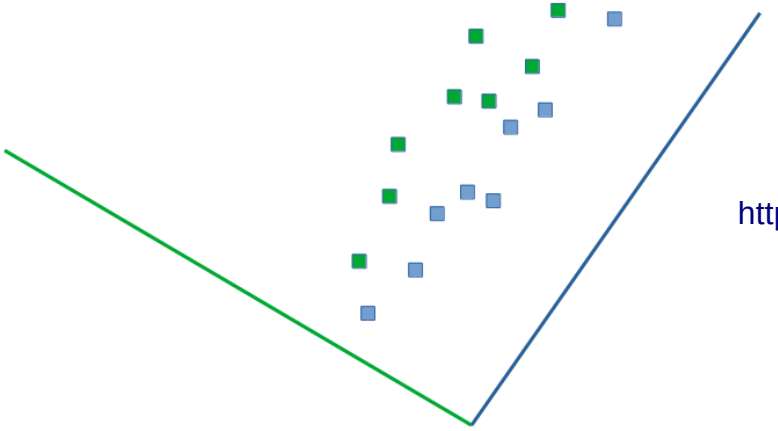
https://math.stackexchange.com/questions/2375598/robust-orientation-of-a-point-cloud



Distances are better preserved when progected on the direction of the greatest variability

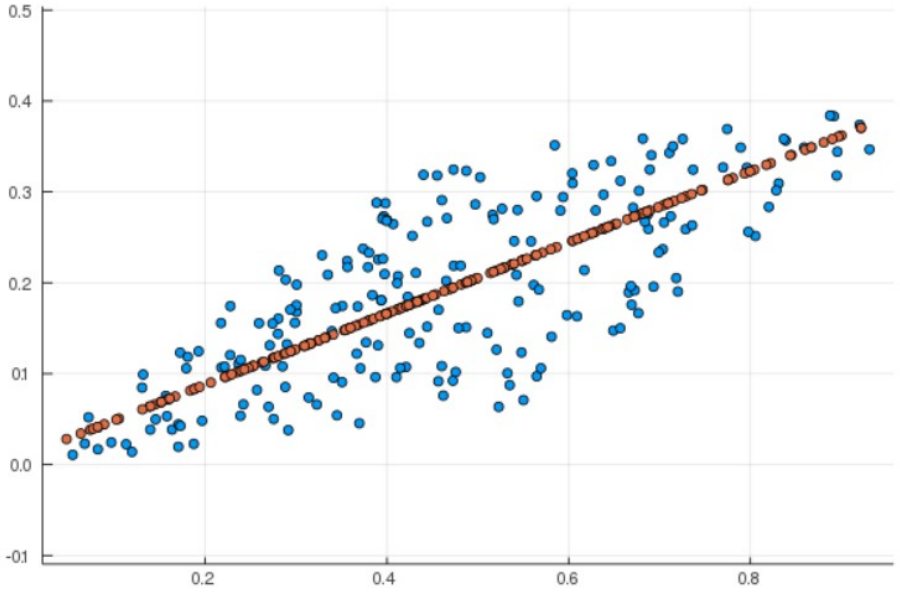
(not always, though)

Data may become not separable after projecting them to the largest variability subspace.



https://www.youtube.com/playlist?list=PLBv09BD7ez_5_yapAg86Od6JeeypkS4YM

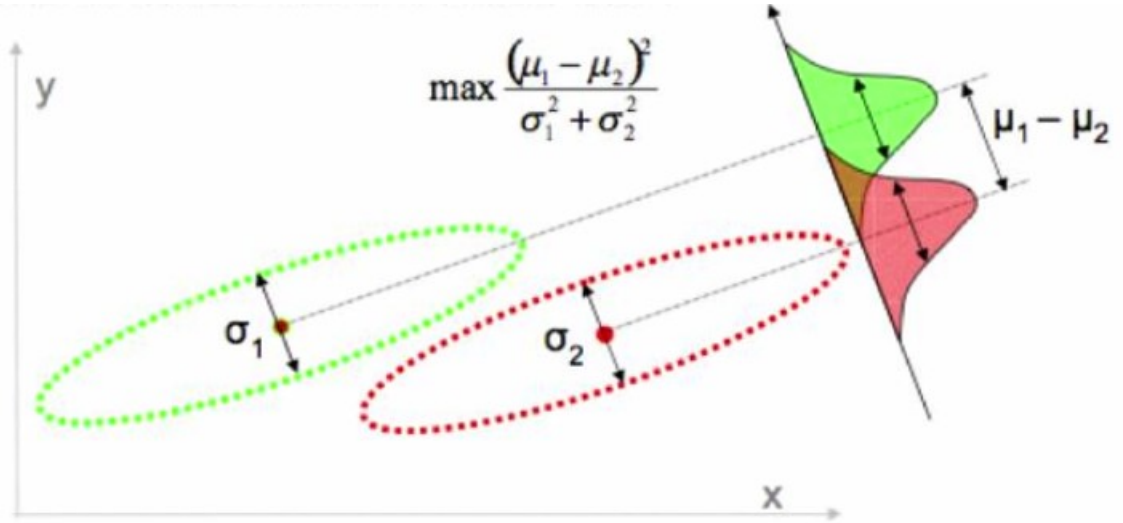
https://stats.stackexchange.com/questions/87198/low-variance-components-in-pca-are-they-really-just-noise-is-there-any-way-to



The results of PCA depend on the scaling of the variables.

PCA creates variables that are linear combinations of the original variables.

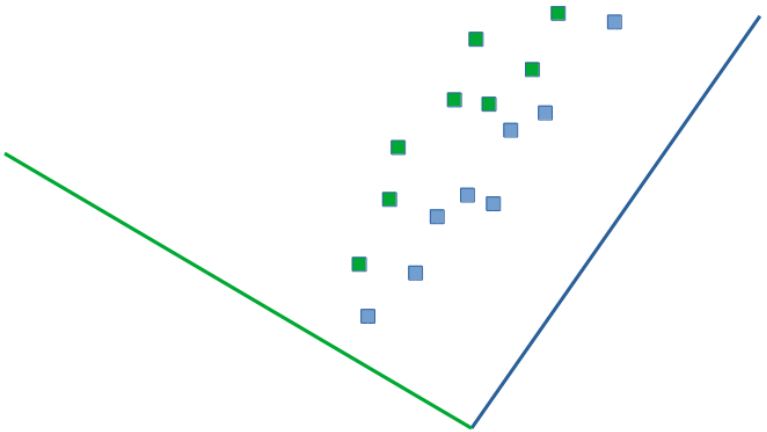
Linear discriminant analysis (LDA)

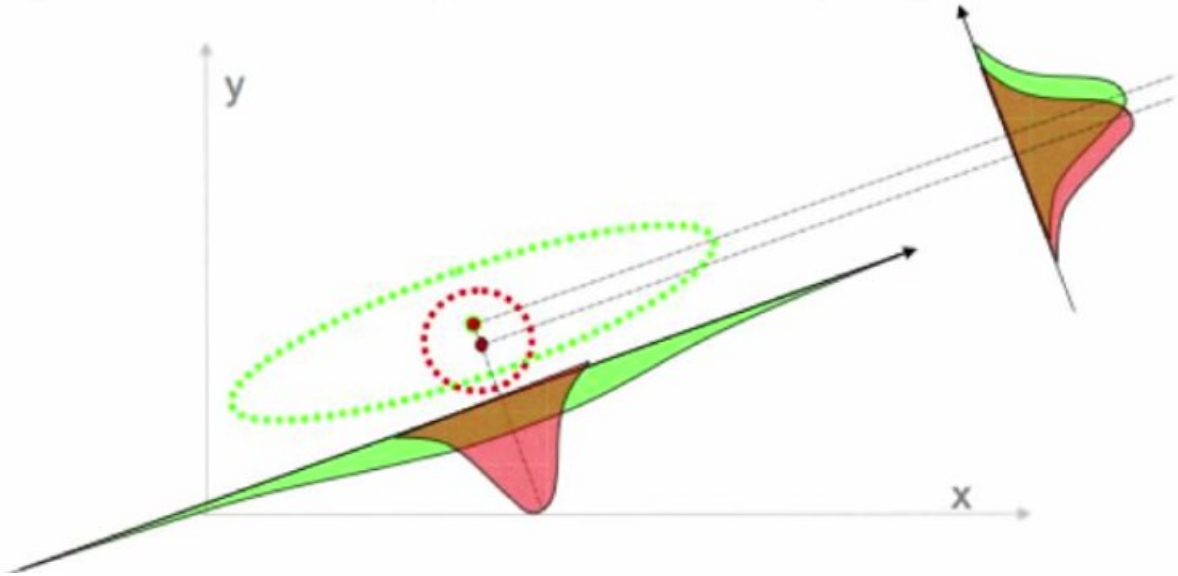


Pick a new direction that gives:

– max separation between the means of projected classes

– min variance within each projected class





https://multivariatestatsjl.readthedocs.io/en/latest/lda.html

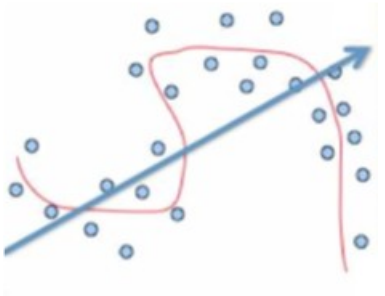
https://multivariatestatsjl.readthedocs.io/en/latest/mclda.html

https://stats.stackexchange.com/questions/169436/how-lda-a-classification-technique-also-serves-as-dimensionality-reduction-tec

PCA assumes that underlying subspace is linear.

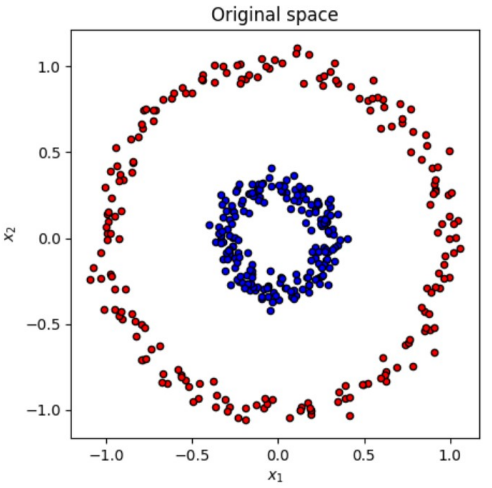
It will find the direction of greatest variability.

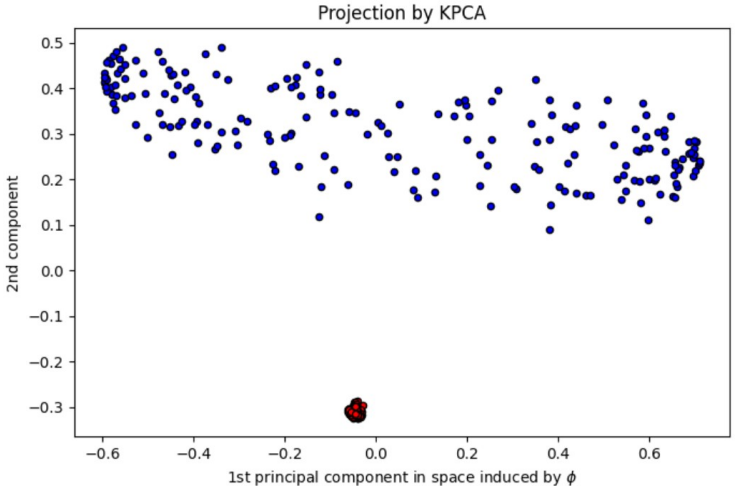
PCA does not perform well when there are nonlinear relationships within the data.



Kernel Principal Component Analysis (kernel PCA)

Using a kernel, the originally linear operations of PCA are performed in a reproducing kernel Hilbert space.





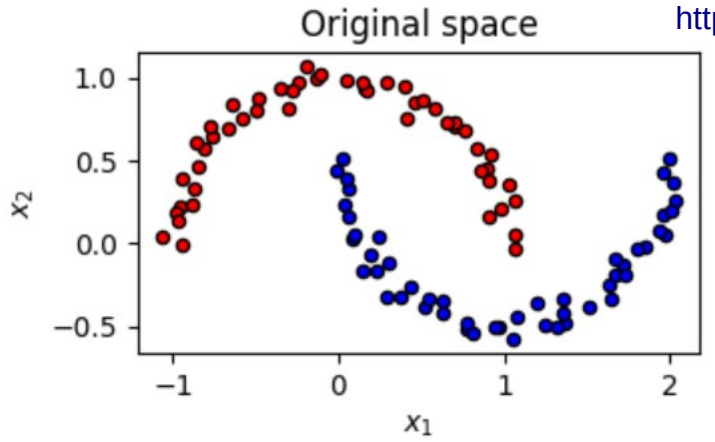
https://scikit-learn.org/stable/auto_examples/decomposition/plot_kernel_pca.html

https://multivariatestatsjl.readthedocs.io/en/latest/kpca.html

https://www.youtube.com/playlist?list=PLBv09BD7ez_5_yapAg86Od6JeeypkS4YM

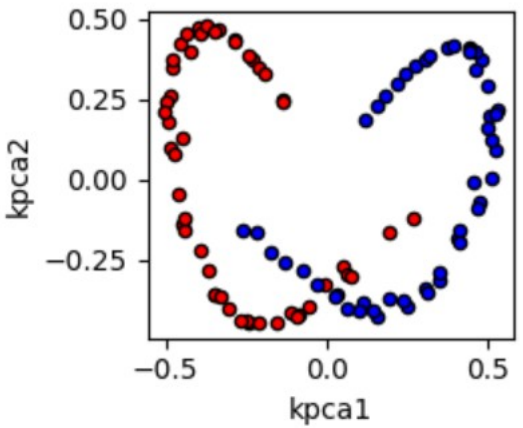
https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.KernelPCA.html

http://scikit-learn.sourceforge.net/dev/modules/metrics.html

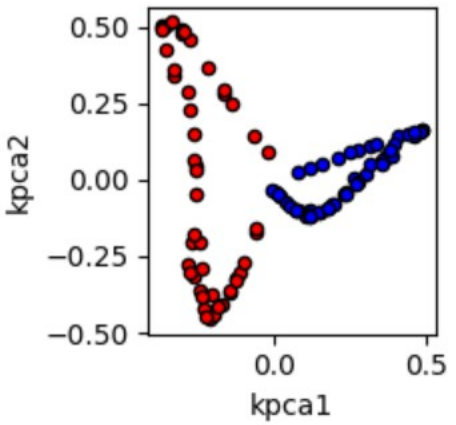


RBF kernel $k(x, y) = \exp(-\gamma \|x - y\|^2)$

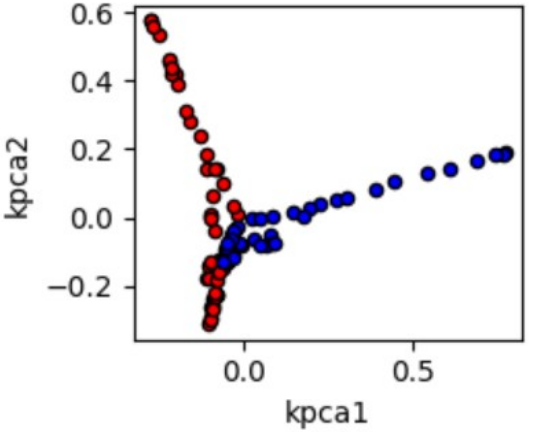
$\gamma = 5$

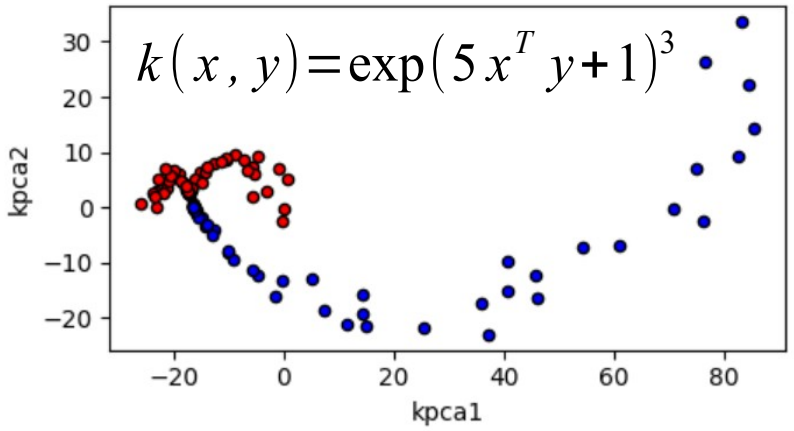


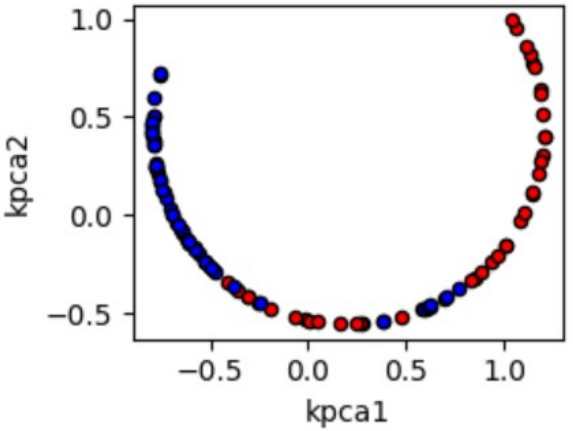
$\gamma = 15$



$\gamma = 35$







$$k(x, y) = \frac{xy^T}{\|x\| \cdot \|y\|}$$

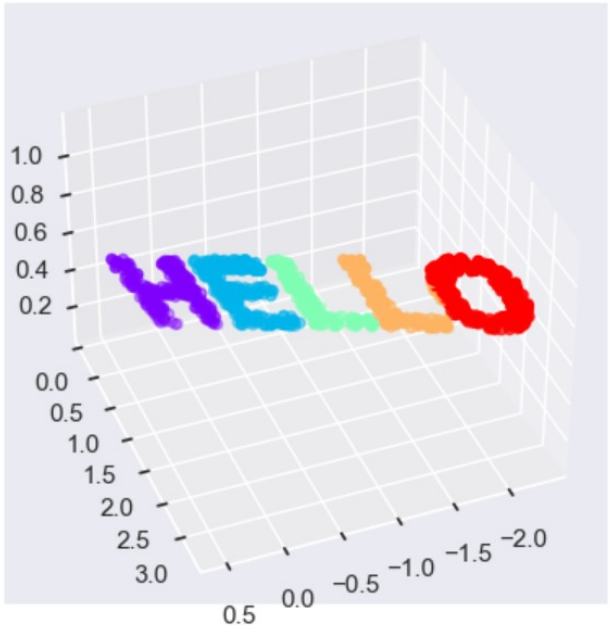
Manifold learning – a class of unsupervised estimators that seeks to describe datasets as low-dimensional manifolds embedded in high-dimensional spaces.

Given high-dimensional embedded data, search for a low-dimensional representation of the data that preserves certain relationships within the data.

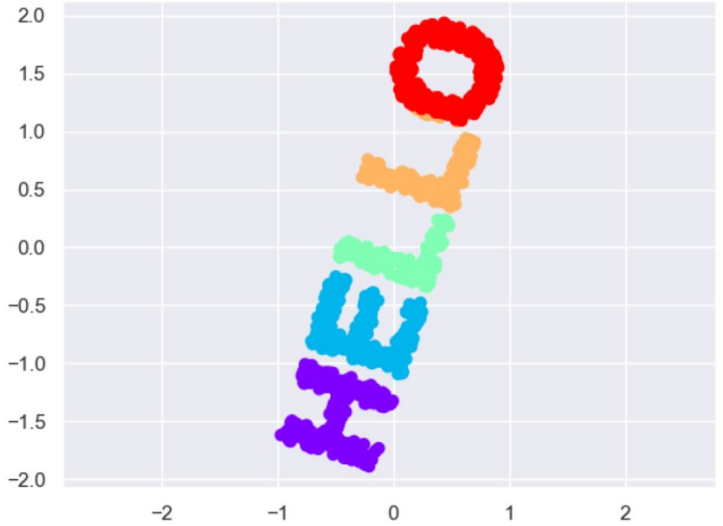
Multidimensional scaling (MDS)

the quantity preserved is the distance between every pair of points

Input: *n*-dimensional data, compute pairwise distances, and then determine the optimal *m*-dimensional embedding to preserve these distances.

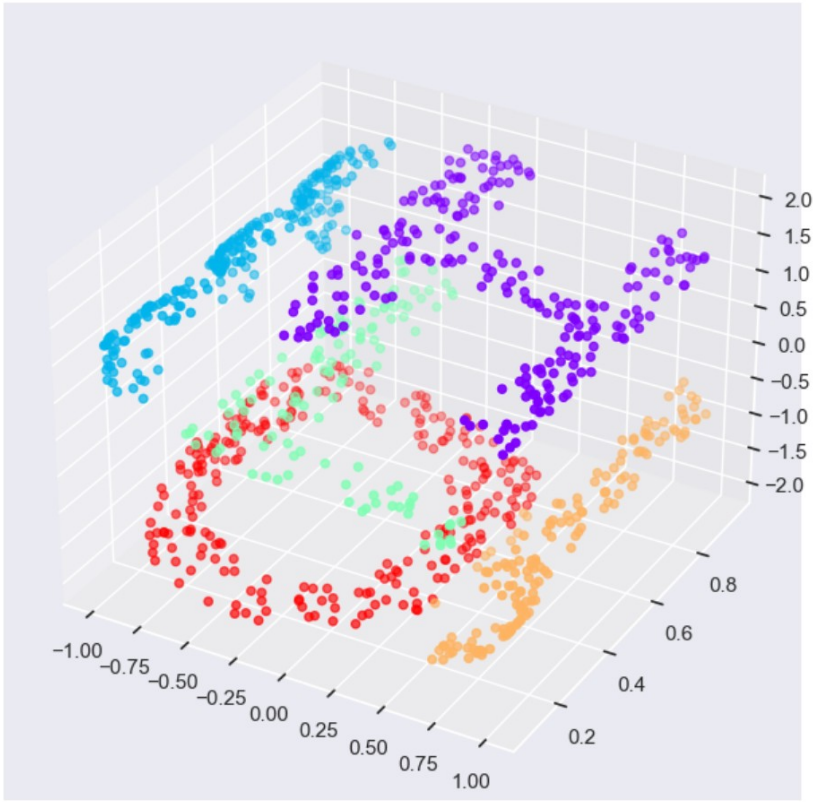


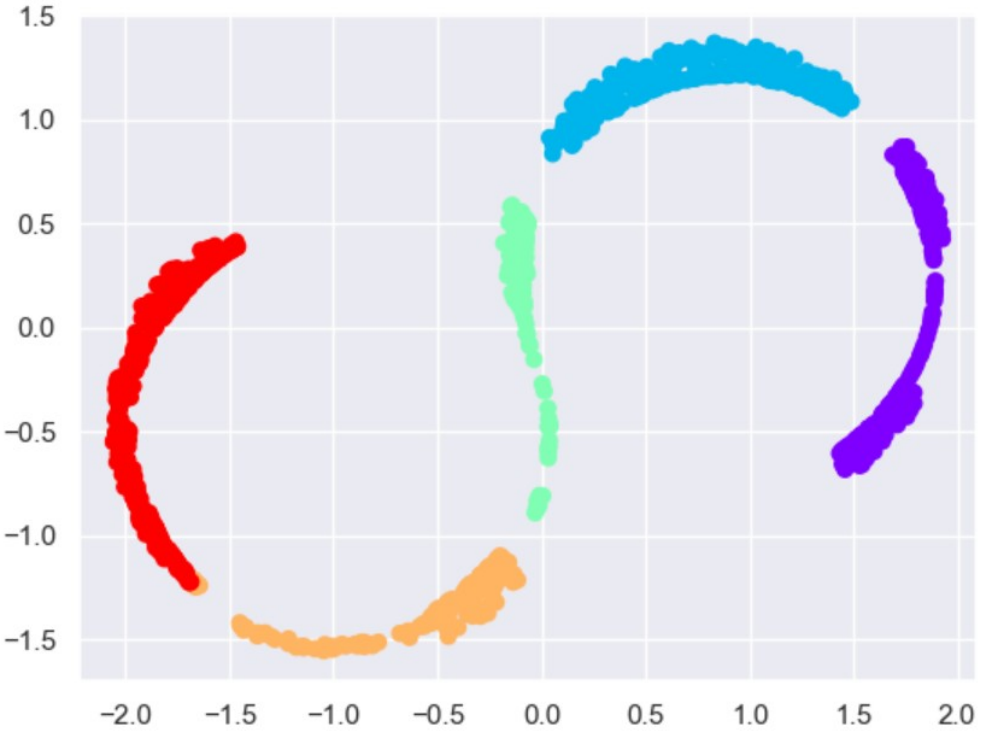
Linear transformation



https://multivariatestatsjl.readthedocs.io/en/latest/cmds.html

https://jakevdp.github.io/PythonDataScienceHandbook/05.10-manifold-learning.html





https://jakevdp.github.io/PythonDataScienceHandbook/05.10-manifold-learning.html

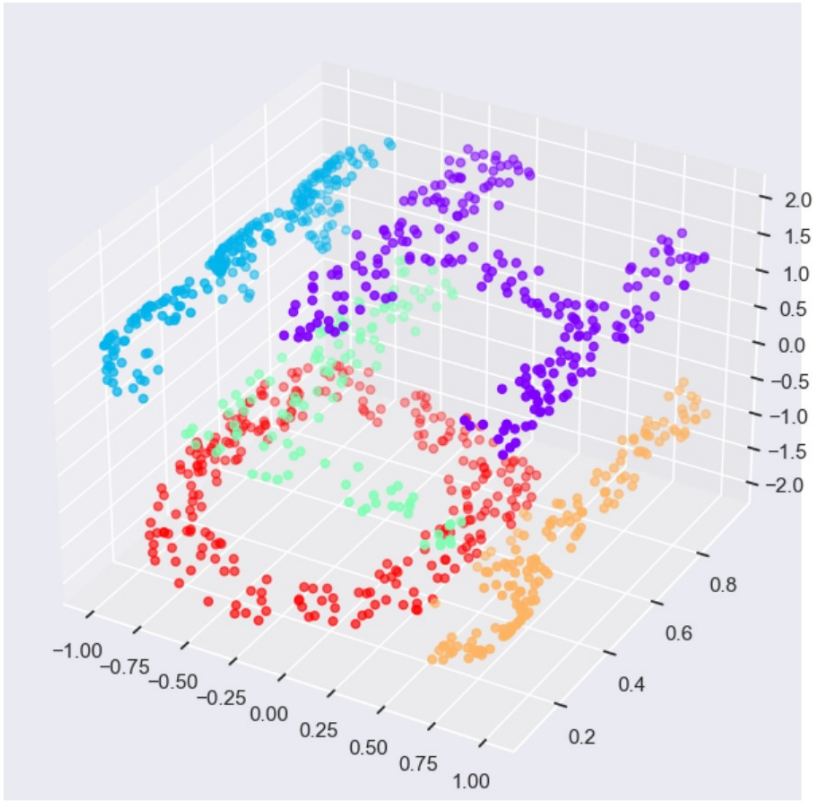
Locally linear embedding (LLE)

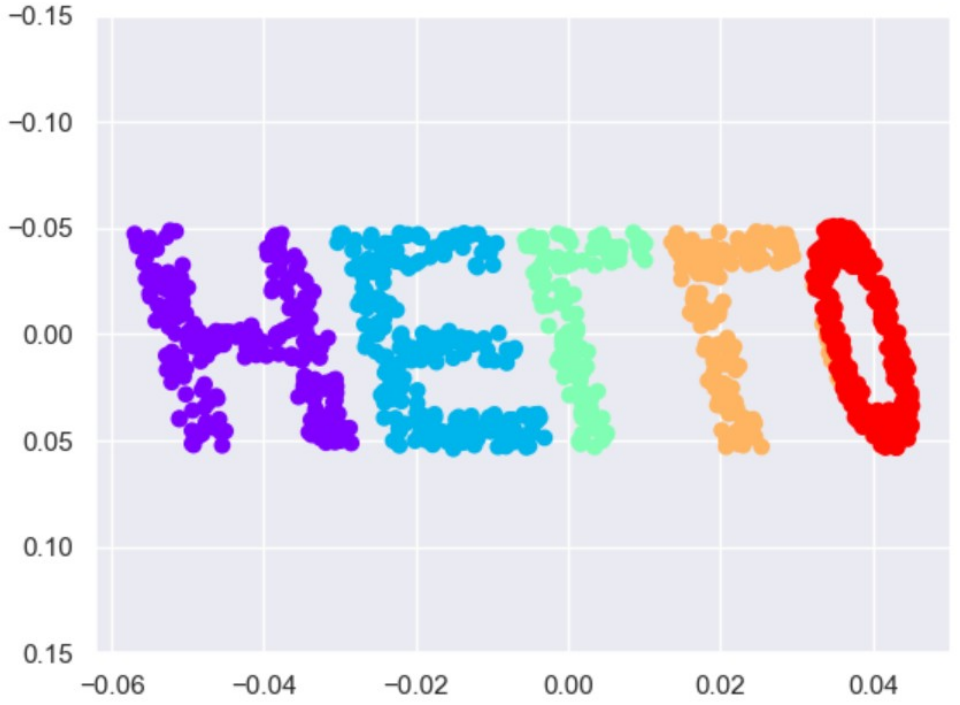
– nonlinear

MDS tries to preserve distances between faraway points when constructing the embedding.

LLE preserves distances only between nearby points

The manifold embedding result is generally highly dependent on the number of neighbors chosen, and there is generally no solid quantitative way to choose an optimal number of neighbors.





https://manifoldlearningjl.readthedocs.io/en/latest/lle.html

https://jakevdp.github.io/PythonDataScienceHandbook/05.10-manifold-learning.html

In non-linear manifolds, the Euclidean metric for distance holds good if and only if neighborhood structure can be approximated as linear.

If neighborhood contains holes, then Euclidean distances can be highly misleading.

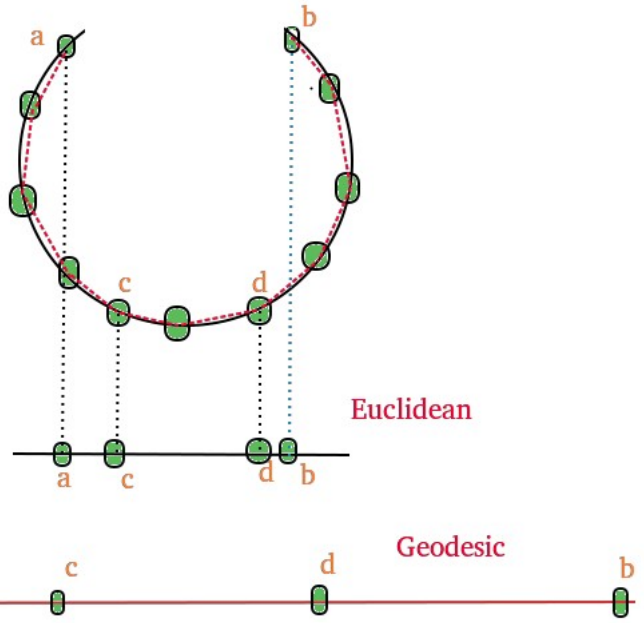
Isometric mapping (IsoMap)

– extends MDS by incorporating the geodesic distances imposed by a weighted graph.

Determine the neighbors of each point.

Construct a neighborhood graph.

Edge length equal to Euclidean distance.



https://blog.paperspace.com/dimension-reduction-with-isomap/

Isomap defines the geodesic distance to be the sum of edge weights along the shortest path between two nodes.

The top *n* eigenvectors of the geodesic distance matrix, represent the coordinates in the new *n*-dimensional Euclidean space.

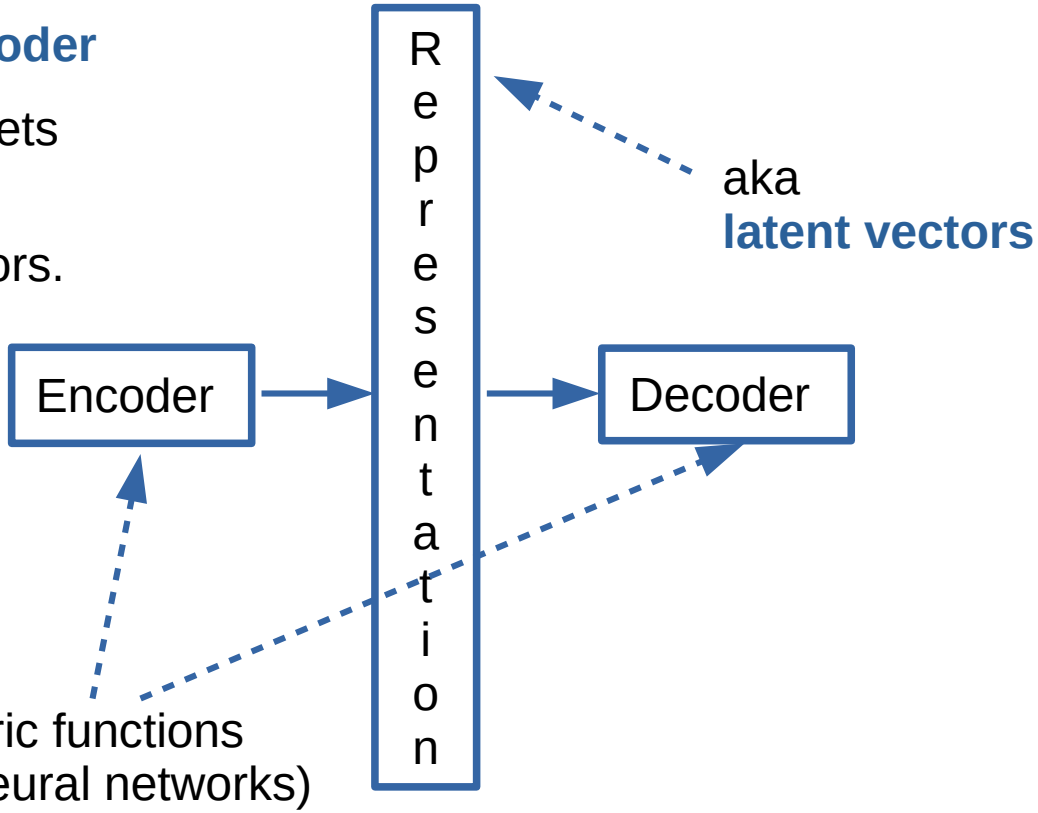
https://manifoldlearningjl.readthedocs.io/en/latest/isomap.html

Autoencoder

– self-supervised learning algorithm that sets the target values to be equal to the inputs.

It is trained to minimise reconstruction errors.

This is a non-linear generalization to PCA.



https://stats.stackexchange.com/questions/120080/whatre-the-differences-between-pca-and-autoencoder

https://arxiv.org/abs/1908.00734

https://arxiv.org/abs/1910.03810

https://arxiv.org/abs/2008.02528

The parameters of the encoding/decoding functions is optimized in order to minimize the reconstruction loss.

By placing constraints on the network, we can discover interesting structure about the data.

Example constraint: the number of hidden units is limited.

=> the network learns a compressed representation of the input.

https://datascience.stackexchange.com/questions/90977/dimensionality-reduction-for-geometric-curves-using-an-autoencoder-what-is-wro

Spectral encoding of categorical features

From the similarity function (or kernel function) construct an adjacency matrix $A_{ij} = K(i, j)$

1-hot encoding step is not necessary here: for the kernel-based ML, only the kernel function between two points matters.

$D = \delta_{ij} \sum_k A_{ik}$ – degree matrix

$L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ – Laplacian

The number of zero eigenvalues correspond to the number of connected components.

If there is only one connected component, we will have only one zero eigenvalue.

Normally it is uninformative and is dropped to prevent multicollinearity of features.

However we can keep it if we are planning to use tree-based models.

The lower eigenvalues correspond to “smooth” eigenvectors.

Keep only these eigenvectors and drop the eigenvectors with higher eigenvalues, because they are more likely represent noise.

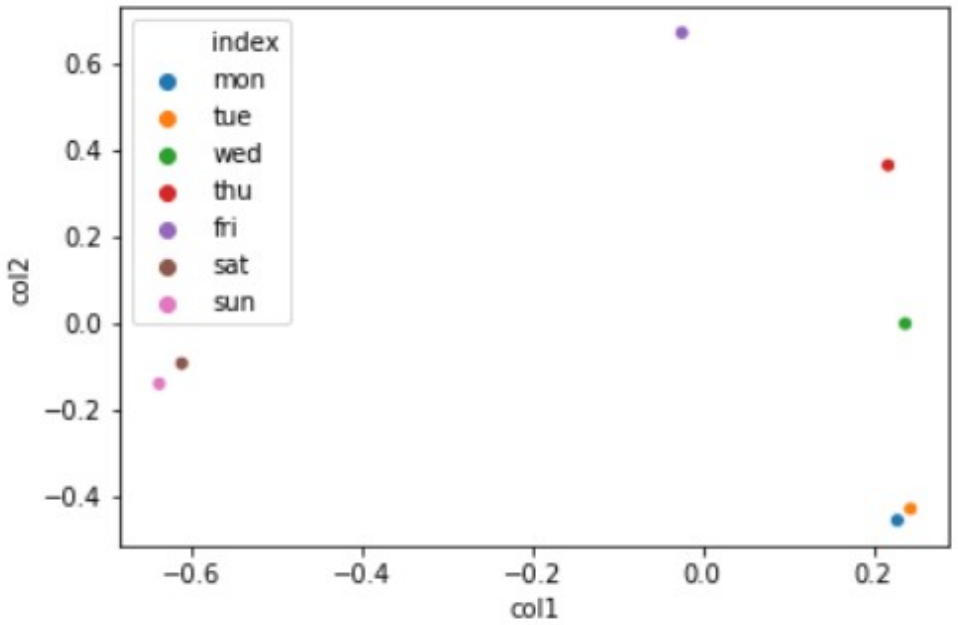
Look for a gap in the matrix spectrum and pick the eigenvalues below the gap.

The resulting truncated eigenvectors can be normalized and represent embeddings of the categorical feature values.

https://towardsdatascience.com/spectral-encoding-of-categorical-features-b4faebdf4a

Example: days of the week

2D spectral encoding




```
using LinearAlgebra

A = [0.0 10.0  9.0  8.0  5.0 2.0  1.0;
     0.0  0.0 10.0  9.0  5.0 2.0  1.0;
     0.0  0.0  0.0 10.0  8.0 2.0  1.0;
     0.0  0.0  0.0  0.0 10.0 2.0  1.0;
     0.0  0.0  0.0  0.0  0.0 5.0  3.0;
     0.0  0.0  0.0  0.0  0.0 0.0 10.0;
     0.0  0.0  0.0  0.0  0.0 0.0  0.0]

A = A + A'

D_isqrt = diagm(0 => vec(sum(A, dims=2) .^(-1/2)))
L = I - D_isqrt * A * D_isqrt
s = eigen(L)
s.values
s.vectors #  the eigenvectors are in the columns
```

$$D = \delta_{ij} \sum_k A_{ik}$$

$$L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

https://towardsdatascience.com/spectral-encoding-of-categorical-features-b4faebdf4a

https://datascience.stackexchange.com/questions/90288/using-the-curse-of-dimensionality-for-encoding-non-ordered-nominal-categorical

Links

Dimensionality Reduction on Statistical Manifolds https://deepblue.lib.umich.edu/handle/2027.42/62266

Information Preserving Embeddings for Discrimination https://ieeexplore.ieee.org/abstract/document/4785953

https://stats.stackexchange.com/questions/23566/functional-principal-component-analysis-fpca-what-is-it-all-about

https://stats.stackexchange.com/questions/28909/pca-when-the-dimensionality-is-greater-than-the-number-of-samples

https://scikit-plot.readthedocs.io/en/stable/decomposition.html

https://stats.stackexchange.com/questions/159705/would-pca-work-for-boolean-binary-data-types

https://stats.stackexchange.com/questions/5774/can-principal-component-analysis-be-applied-to-datasets-containing-a-mix-of-cont