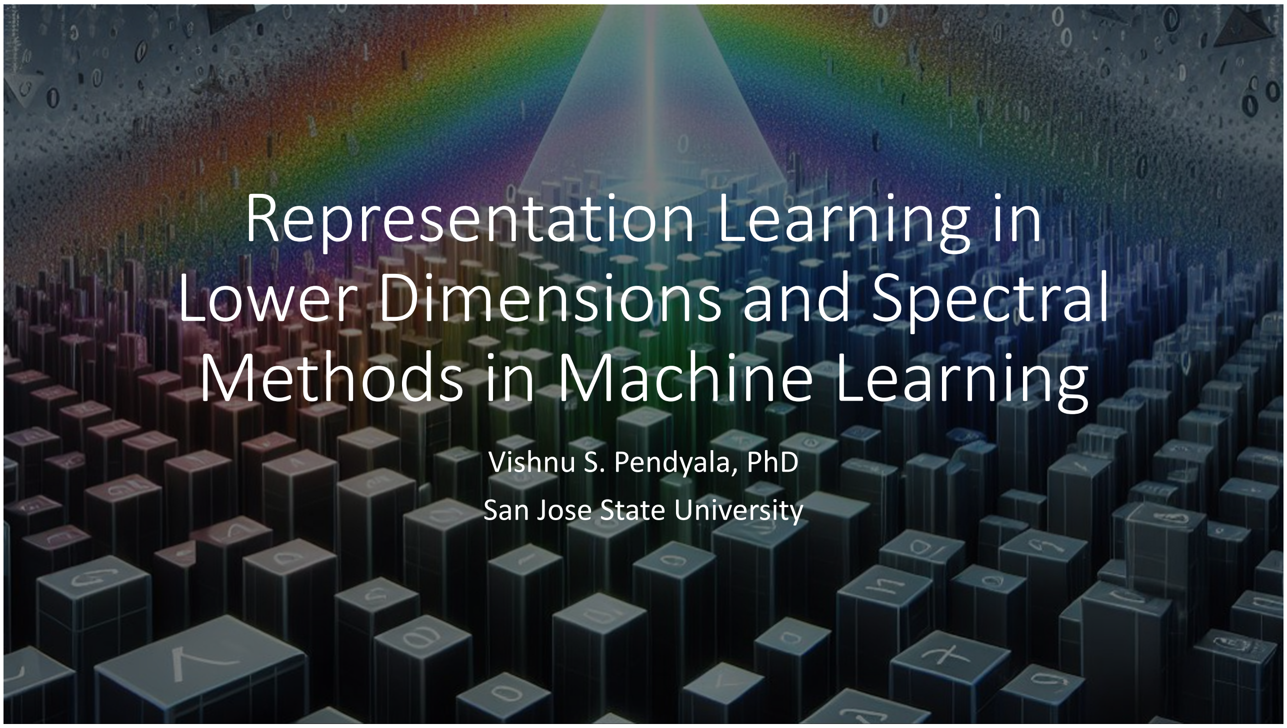
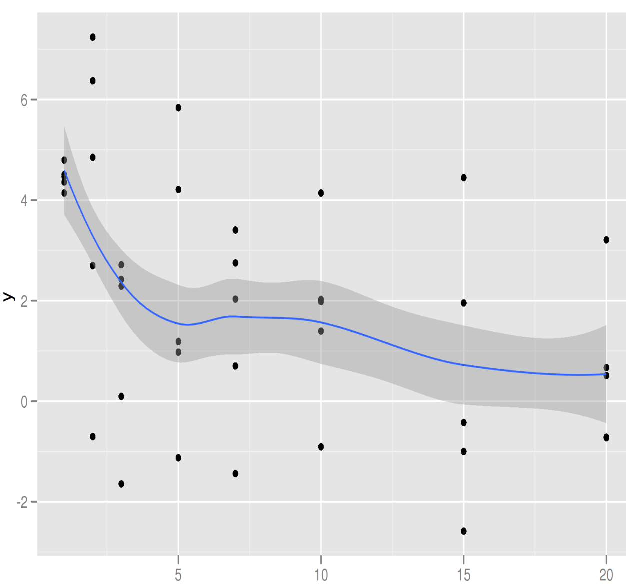


Representation Learning in Lower Dimensions and Spectral Methods in Machine Learning

Vishnu S. Pendyala, PhD
San Jose State University

Which of these reveal the hidden structure of data?

0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
0.9953	0.9955	0.9956	0.9957	0.9959	0.9960	0.9961	0.9962	0.9963	0.9964
0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9979	0.9980	0.9981
0.9981	0.9982	0.9982	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986
0.9987	0.9987	0.9987	0.9988	0.9988	0.9989	0.9989	0.9989	0.9990	0.9990



Why learn representations in lower dimensions?

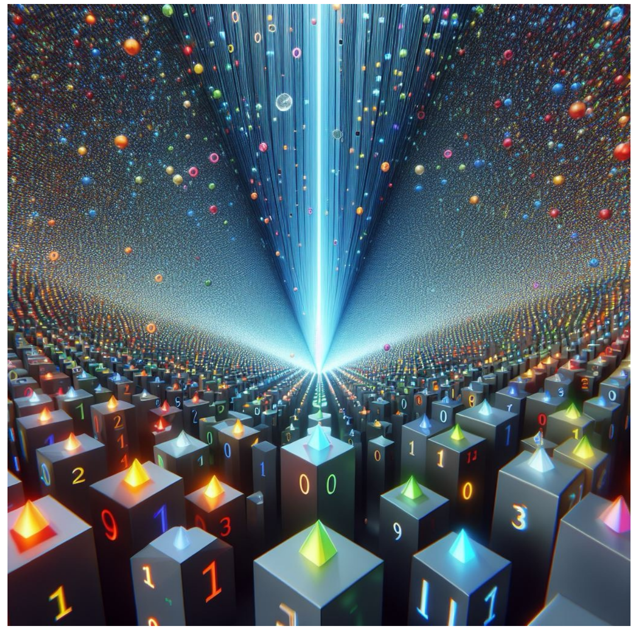
- VISUALIZATION: PROJECTING DATA POINTS ONTO A 2D OR 3D SPACE CAN HELP IN IDENTIFYING PATTERNS, CLUSTERS, OR OUTLIERS.
- DIMENSIONALITY REDUCTION CAN SIMPLIFY THE DATA AND MAKE IT MORE MANAGEABLE.
- COMPUTATIONAL EFFICIENCY, ESPECIALLY FOR ML ALGORITHMS BASED ON DISTANCE CALCULATIONS CAN LEAD TO FASTER TRAINING AND INFERENCE TIMES.
- OVERFITTING MITIGATION BY REDUCING THE COMPLEXITY OF THE MODEL.
- LOWER-DIMENSIONAL REPRESENTATIONS CAN BE MORE INTERPRETABLE WHEN TRYING TO UNDERSTAND AND COMMUNICATE THE RESULTS OF A MACHINE LEARNING MODEL.

More reasons to learn representations in lower dimensions

- Data Compression beneficial for storage and transmission of data.
- Makes it easier for clustering and classification algorithms to identify patterns and group data points.
- Anomalies or outliers may become more apparent in lower-dimensional space, aiding in the detection of unusual or suspicious data points.
- Identifying and retaining the most important features of the data while discarding less important or redundant ones can lead to more efficient and effective models.
- Reduces the impact of noise or random variations in the data, making it easier for models to focus on the underlying patterns.

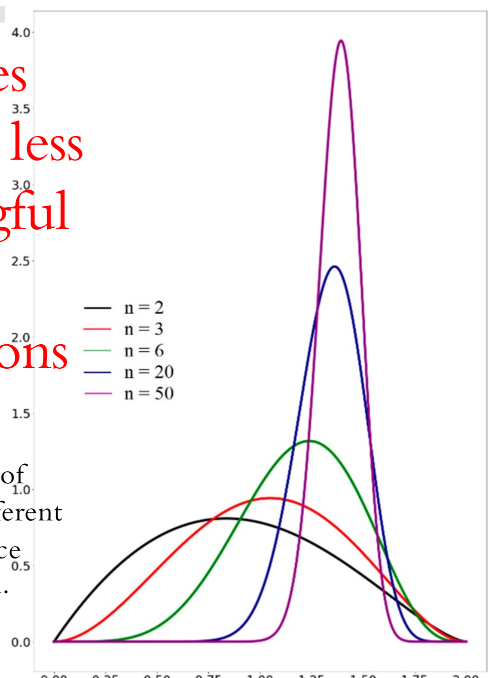
Training data needs grow exponentially with dimensions

=> Learning in high dimensions is intractable



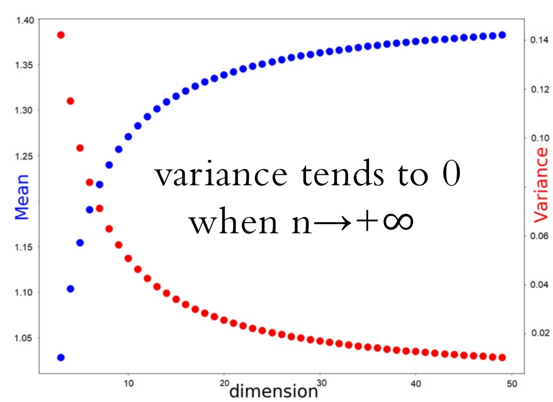
Distances become less meaningful in high dimensions

Distribution of distances for different values of space dimension n.



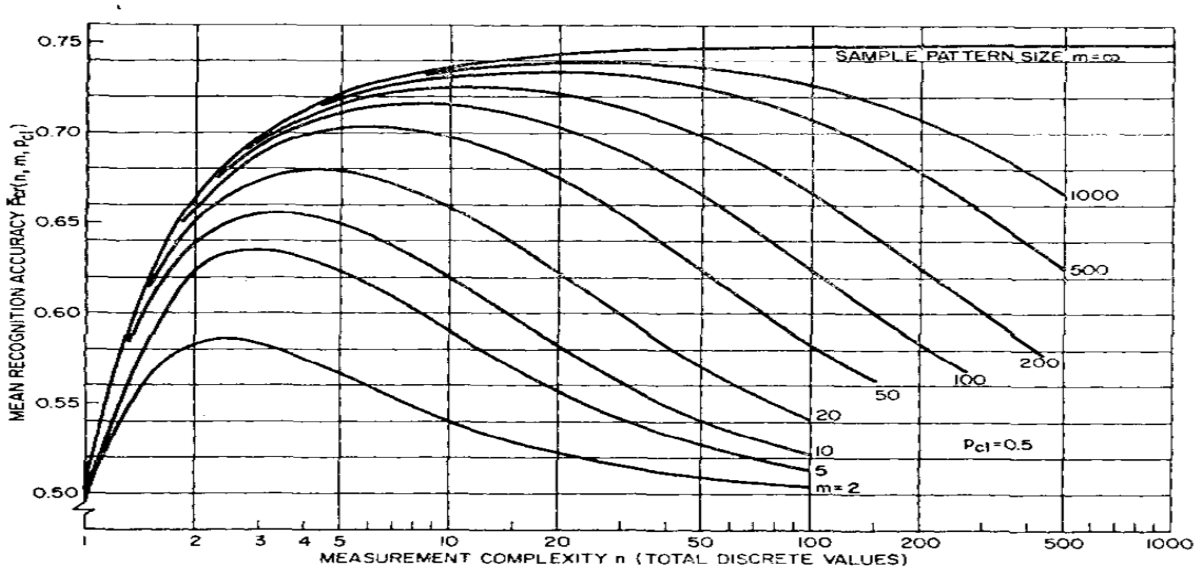
(a)

Lellouche, S., & Souris, M. (2019). Distribution of distances between elements in a compact set. *Stats*, *3*(1), 1-15.



(b)

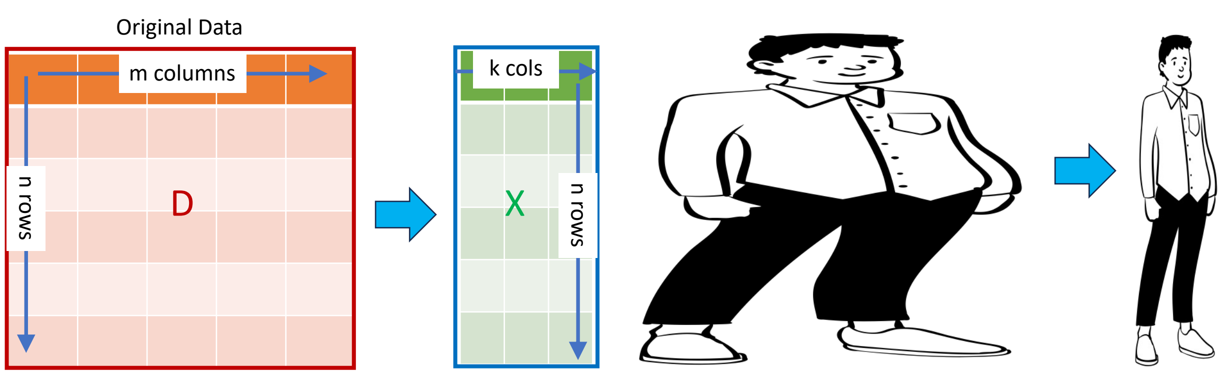
Curse of dimensionality: Hughes phenomenon



Source: Hughes, G. F. (1968). On the mean accuracy of statistical pattern recognizers. IEEE Transactions on Information Theory, 14(1), 55-63.

Principal Component Analysis (PCA)

- Does not require labels – unsupervised
- Data with many (m) features abstracted / condensed into fewer (k) principal components (k < m) that are synthetic



More Intuition

Movies are shot in 3D but we watch in 2D without much loss of information even when dropping the 3rd dimension.

We listen to music when working. But when we need to focus, we drop the music dimension without forgoing much.

Class Survey: Rate these on a scale of 1-5

- Everyone in my team contributes to my learning
- My team provides great insights during discussions
- Each team member creates a positive environment

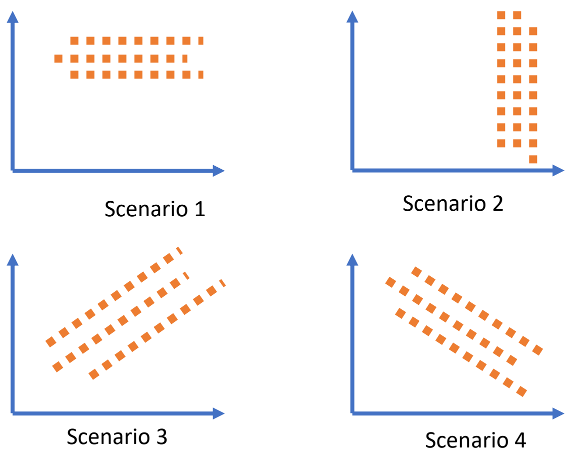
Redundancy

- Can be replaced by

- My team generates great synergy

Latent intent

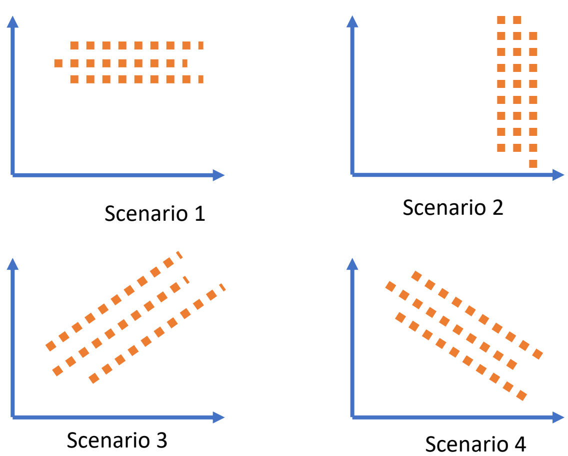
Which of the 4 plots is most likely?



X-axis: DATA 245 Score
Y-axis: DATA 228 Score

Each vector represents the scores obtained by a student in the MSDA program

If only one subject's grade is available for hiring an RA, what would that subject be?

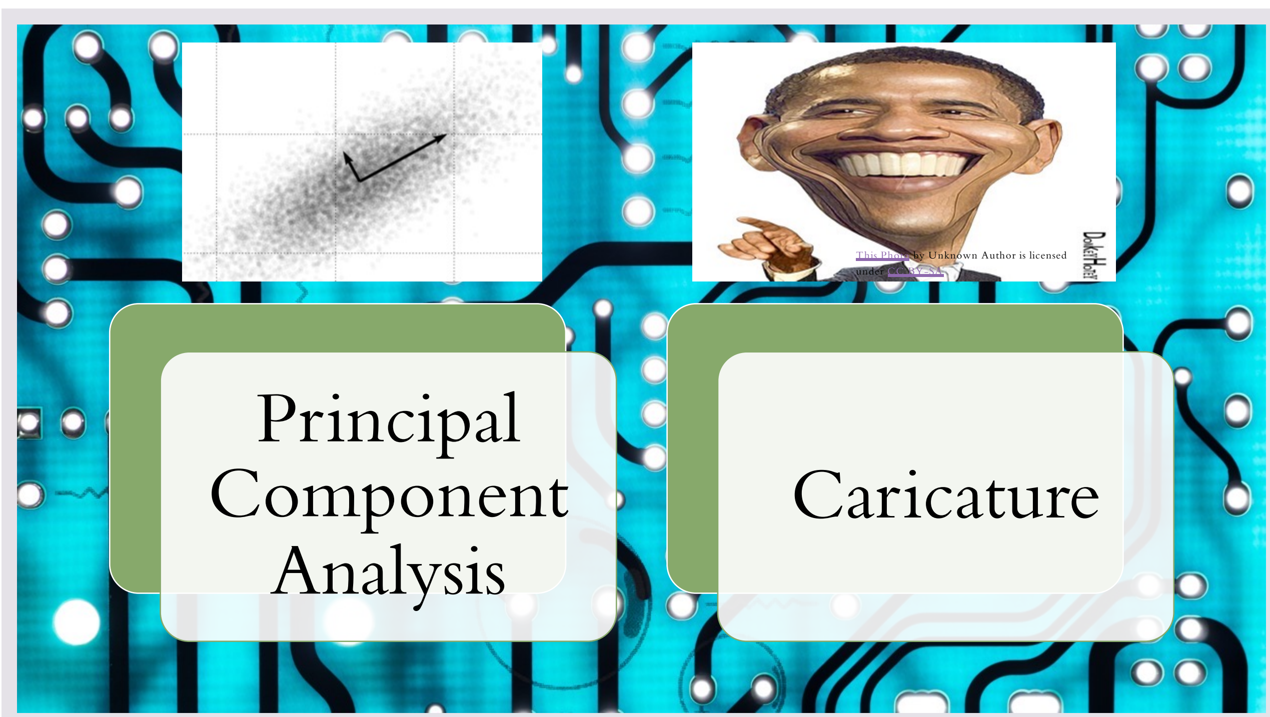


X-axis: DATA 245 Score
Y-axis: DATA 228 Score

Each vector represents the scores obtained by a student in the MSDA program

It is the variance that matters!


Source: https://www.flickr.com/photos/grantp/668054637


Principal Component Analysis
Caricature

How do we capture the oddities or unique variations in a dataset?

Principal Component Analysis (PCA) – A Linear Technique

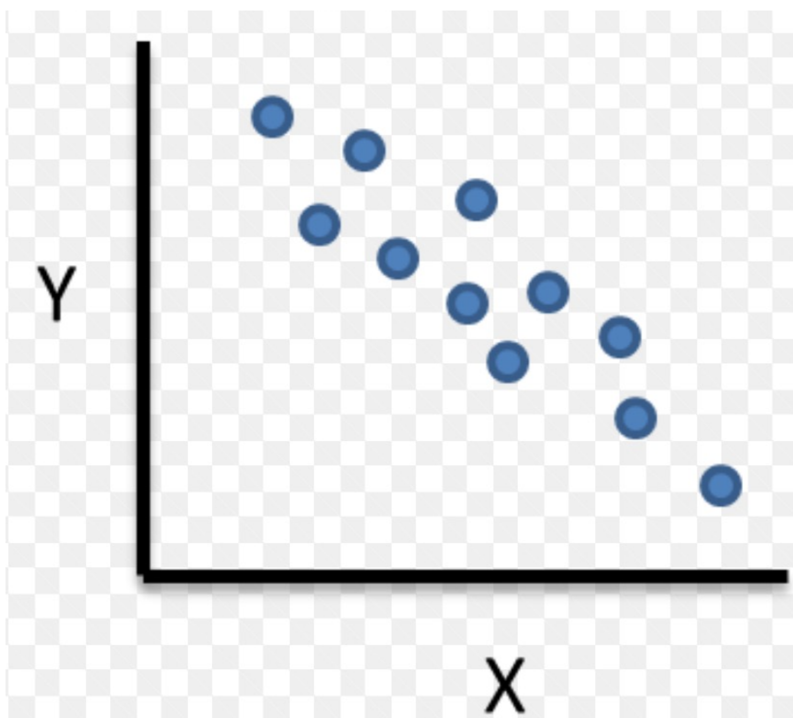
Originated in: *Pearson, K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space. Philosophical Magazine, 2, 559-572.*

Significant Improvement: *Hotelling, H. (1933). Analysis of a Complex of Statistical Variables into Principal Components. Journal of Educational Psychology, 24(6), 417-441.*

First application: *Goodall, D. W. (1954). Objective methods for the classification of vegetation. III. An essay in the use of factor analysis. Australian Journal of Botany, 2(3), 304-324.*

Can we use just one dimension for this 2D data?

What’s the statistical parameter that is indicative of the direction?



PCA: Ideas

Capture as much important information in a dataset, but using fewer features (dimensions)

Drop least important features so generated; importance is indicative of the variance in the data the new feature captures

Keywords: variance; orthogonal; relative importance

Transform the feature space: From correlated features to entirely uncorrelated, orthogonal, synthetic features

Result: Original dataset is now projected in a fewer dimensional space

Think: covariance matrix; eigen vectors (orthogonal for symmetric matrices like the covariance matrix); eigen values

The notion of variance and correlation

- One variable, Var(x): $\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{n}$ Two Variables, Cov(x,y): $\frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{n}$

x and y are independent => cov(x,y) = 0; directly correlated => cov(x,y) > 0
inversely correlated => cov(x,y) < 0

Correlation Coefficient = $\frac{cov(x,y)}{\sqrt{Var(x).Var(y)}}$

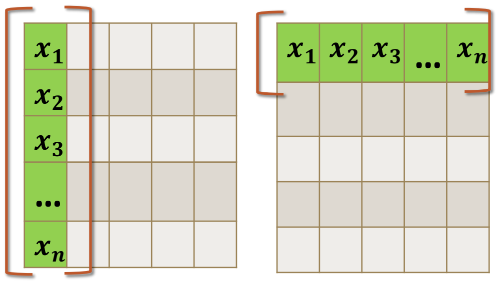
- Three or more variables, Covariance Matrix:

Cov(x,x) = var(x)	Cov(x,y)	Cov(x,z)
Cov(y,x)	Cov(y,y) = var(y)	Cov(y,z)
Cov(z,x)	Cov(z,y)	Cov(z,z) = var(z)

If the data is already mean adjusted, what is the covariance?

Kernel Matrix vs Covariance Matrix

$X^T=$ $\quad$ $X=$



$X^TX=$Kernel Matrix

$\langle x_1, x_1\rangle$	$\langle x_1, x_2\rangle$	$\langle x_1, x_3\rangle$	...	$\langle x_1, x_n\rangle$
$\langle x_2, x_1\rangle$	$\langle x_2, x_2\rangle$	...	...	...
$\langle x_3, x_1\rangle$	...	$\langle x_3, x_3\rangle$	...	...
...	...	...	...	...
$\langle x_n, x_1\rangle$	...	...	...	$\langle x_n, x_n\rangle$

Corollary: What is XX^T ?

Eigenvalues and Eigenvectors

Eigenvalues (λ) are scalar values that represent how a linear transformation (represented by a matrix) stretches or compresses space.

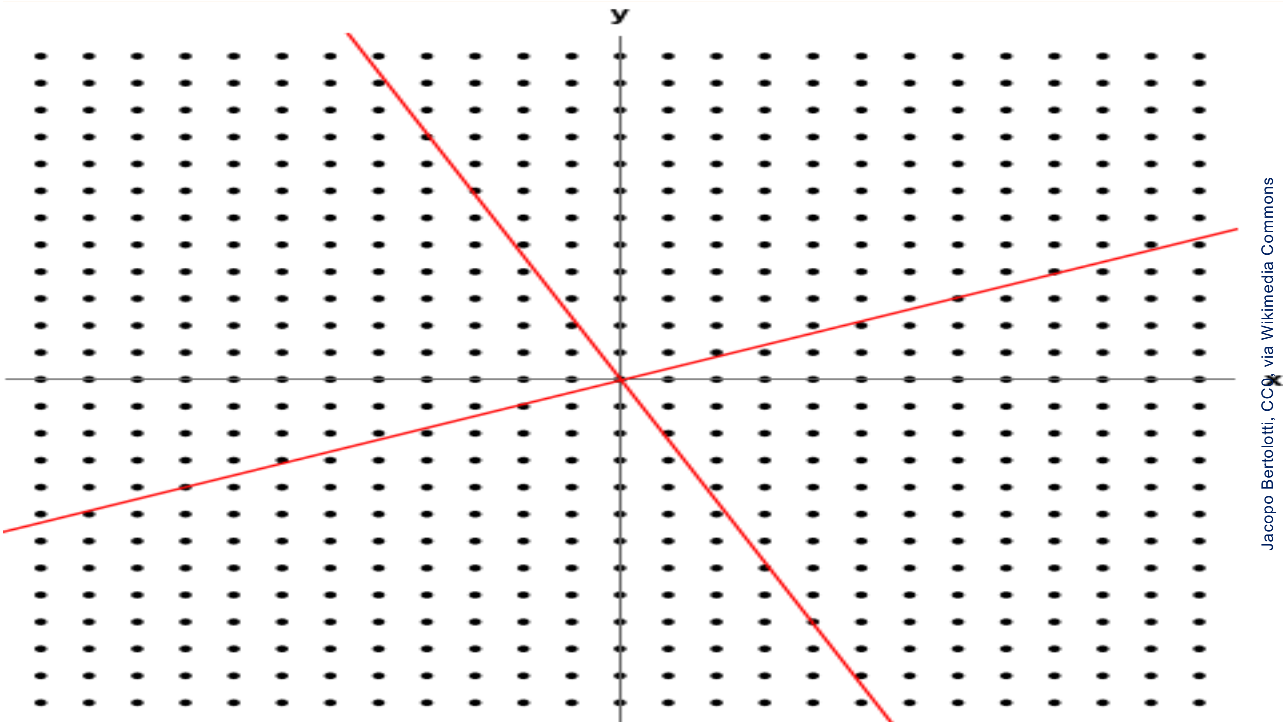
For a given eigenvalue, there may be multiple eigenvectors. The set of all eigenvectors corresponding to a particular eigenvalue is called the eigenspace.

If A is a square matrix, λ is an eigenvalue of A if there exists a non-zero eigenvector v such that $Av = \lambda v$.

Eigenvectors are non-zero vectors that only change by a scalar factor, λ (direction does not change) when a linear transformation, A is applied

Two properties useful for PCA:

- Orthogonality: Eigenvectors corresponding to distinct eigenvalues are orthogonal (linearly independent) for symmetric matrices
- Eigenbasis: Eigenvectors can form a basis for a vector space



Finding Eigenvalues and Eigenvectors

- Characteristic equation: Determinant of (A - λI) = 0, where A is the matrix, λ is the eigenvalue, and I is the identity matrix of the same size as A.
- Why? Ax = λx => (Ax – λx) = 0 => (A – λI)x = 0 and by definition, eigenvectors are non-zero, so det(A – λI) = 0
- Solve the characteristic equation for λ to find the eigenvalues.
- To find the eigenvectors, for each eigenvalue λ:
  - Substitute λ back into the equation (A - λI)v = 0, where v is the eigenvector.
  - Solve the resulting system of linear equations to find the eigenvector v.

Finding eigenvectors: 2x2 matrix example

- Matrix A: $\begin{bmatrix} 4 & 2 \\ 3 & -1 \end{bmatrix}$
- Characteristic equation: : $\begin{bmatrix} 4-\lambda & 2 \\ 3 & -1-\lambda \end{bmatrix}$ = 0

=> (4 - λ)(-1 - λ) - 6 = 0

λ^2 - 3λ - 10 = 0 => (λ - 5)(λ + 2) = 0

- Eigenvalues: λ = 5, -2
- **Eigenvector for λ = 5:**
- (A - 5I)v = 0
- $\begin{bmatrix} -1 & 2 \\ 3 & -6 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = 0$
- => -x + 2y = 0 and 3x – 6y = 0
- Both the equations are identical to x = 2y
- Solving this system gives many eigenvectors such as v = [2, 1].
- **Eigenvector for λ = -2:**
- (A + 2I)v = 0
- $\begin{bmatrix} 6 & 2 \\ 3 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = 0$
- Again, the resulting equations are identical to y = -3x
- Many eigenvectors like [1, -3], [2, -6]

Finding eigenvectors: 3x3 symmetric matrix

- Matrix A: $\begin{bmatrix} 2 & 1 & 3 \\ 1 & 2 & 3 \\ 3 & 3 & 20 \end{bmatrix}$
- Characteristic equation:

$\begin{bmatrix} 2-\lambda & 1 & 3 \\ 1 & 2-\lambda & 3 \\ 3 & 3 & 20-\lambda \end{bmatrix}$ = 0

Expand the determinant using cofactor expansion:

Choose a row or column to expand along. For this example, let's expand along the first row:

(2 - λ) * (submatrix determinant) - 1 * (submatrix determinant) + 3 * (submatrix determinant) = 0

Evaluate each 2x2 submatrix determinant:

(2 - λ)[(2 - λ)(20 - λ) - 9] - 1[(1)(20 - λ) - 9] + 3[(1)(3) - (2 - λ)(3)]

=> λ^3 - 25λ^2 + 114λ - 120 = 0

=> (λ - 1)(λ - 5)(λ - 20) = 0

λ = 1, 5, 20

Eigenvectors can be found like before

What if the eigen values cannot be found or are not real numbers?

Not Possible for Covariance matrices!

Eigen vectors of a Symmetric Matrix

- If all the values in the symmetric matrix ($S^T = S$) are real, then eigen values and eigen vectors are real numbers as well.
- Eigen vectors of a real symmetric matrix are orthogonal
- Spectral decomposition: $S = Q\Lambda Q^T$ where Q is an orthogonal matrix
- Q's columns are eigen vectors of S
- Λ is a diagonal matrix of eigen values of S
- All the eigen values of a positive definite symmetric matrix are positive

28

Why are the eigenvalues of the Gram matrices nonnegative?

Let λ be any eigenvalue of K = $X^T X$ and the corresponding eigen vector, v. Then,

$$(X^T X)v = \lambda v$$
$$v^T X^T X v = v^T \lambda v$$
$$(Xv)^T (Xv) = \lambda v^T v$$
$$(Xv)^T (Xv) \; and \; v^T v \; are \; both \; norms$$
$$so \; have \; to \; be \; positive$$
$$\left||Xv|\right|^2 = \lambda\left||v|\right|^2$$
$$\Rightarrow \lambda > 0$$

29

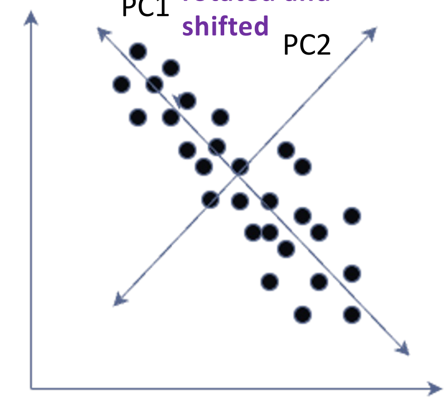
Other interesting properties

- The eigenvalues of the covariance matrix, XX^T can also be similarly proved to be > 0
- The Gram matrices XX^T and X^TX share the same nonzero eigenvalues (why?).

$$X^T X u = \lambda u$$
$$XX^T X u = X\lambda u$$
$$XX^T (Xu) = \lambda (Xu)$$
$$XX^T \tilde{u} = \lambda \tilde{u}$$

Principal Components

- Sort the absolute |eigenvalues|
- PC1: Direction of maximum spread (variance) – the direction of the 1st eigenvector corresponding to the largest absolute |eigenvalue|
- PC2: 2nd eigenvector direction covering maximum residual variation left in the data, orthogonal to PC1
- PC3 (if the original data has 3+ dimensions): 3rd eigenvector direction with maximum spread left in data after PC1 and PC2, orthogonal to PC1 and PC2
- So on and so forth
- Observation: PC1 covers most the spread; PC2 is almost redundant
- PC2 can be dropped without losing significant information that the data conveys



Equations of the new axes in terms of the old

- Original features: $x_1, x_2, x_3, \ldots x_m$ New uncorrelated PCs: $z_1, z_2, z_3, \ldots, z_m$
- PC_1 (*direction of the most spread*): $z_1 = k_{11}x_1 + k_{12}x_2 + \cdots + k_{1m}x_m$
- $[\boldsymbol{k_{11}}, \boldsymbol{k_{12}}, \boldsymbol{k_{13}}, \ldots, \boldsymbol{k_{1m}}]$ *is the 1st eigenvector of the covariance matrix*
- PC_2 (*direction of* 2^{nd} *eigenvector*): $z_2 = k_{21}x_1 + k_{22}x_2 + \cdots + k_{2m}x_m$
- $[\boldsymbol{k_{21}}, \boldsymbol{k_{22}}, \boldsymbol{k_{23}}, \ldots, \boldsymbol{k_{2m}}]$ *is the 2nd eigenvector of the covariance matrix*
- …
- PC_m: $z_m = k_{m1}x_1 + k_{m2}x_2 + \cdots + k_{mm}x_m$
- The PCs capture **all** of the information in the original features
- Usually, the 1st few PCs capture most of the information; rest are redundant
- If so, keep the 1st few and drop the rest => dimensionality reduction

PCA: Dimensionality Reduction

- Cattell, Raymond B. (1966). "The Scree Test For The Number Of Factors". Multivariate Behavioral Research. 1 (2): 245–276.
- Scree Test: procedure of finding statistically significant factors
- Y-axis: eigenvalues represent the percentage of variance explained by each PC

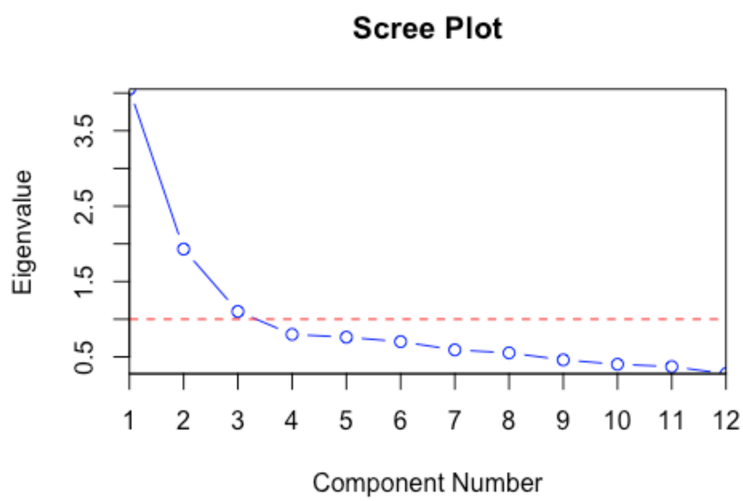
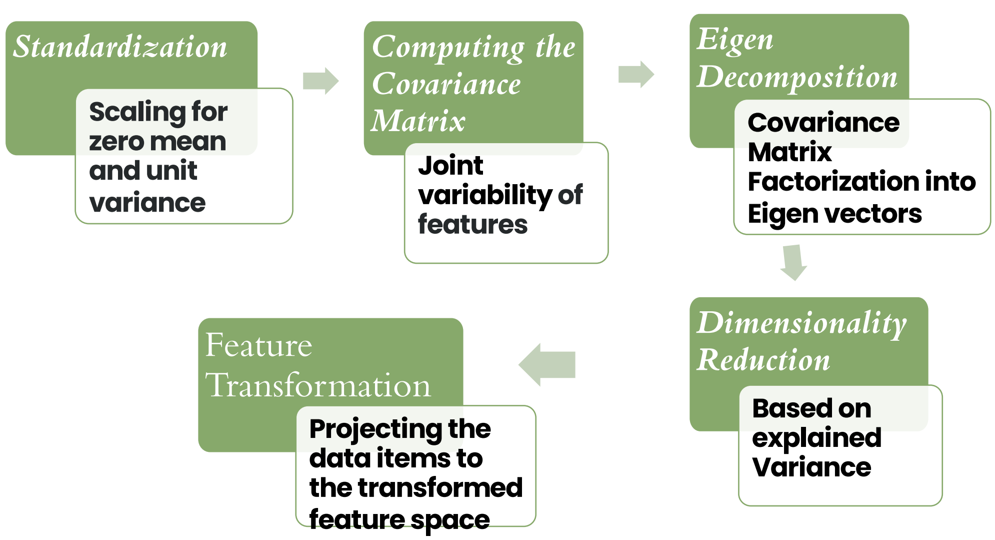


Figure By Staticshakedown - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=75715167

PCA – The Process



PCA: Implementation

- Data (2 features, 5 observations):
- X = [[1, 2], [2, 3], [4, 5], [5, 7], [8, 9]]

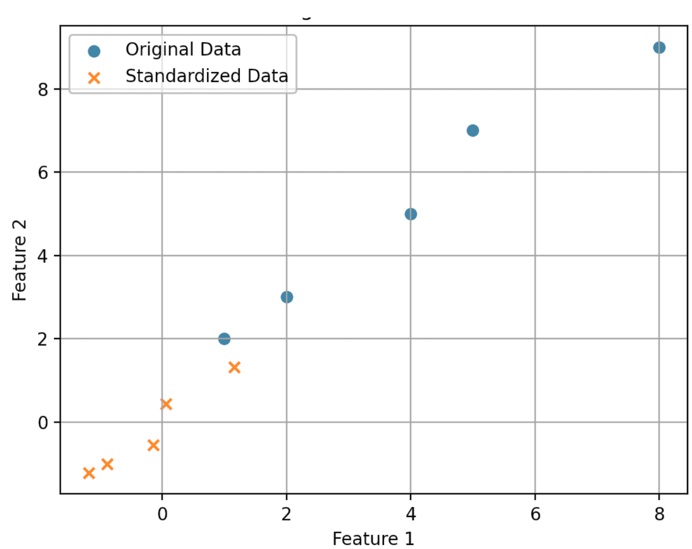
- Standardization: Calculate the mean and standard deviation for each variable:
  - mean_X = np.mean(X, axis=0)
    # [4.0, 5.2]
    # mean_X1 = (1 + 2 + 4 + 5 + 8) / 5 = 4.0
    # mean_X2 = (2 + 3 + 5 + 7 + 9) / 5 = 5.2

- std_X = np.std(X, axis=0)
  - # [2.44948974 2.56124969]

- Z = (X - mean_X) / std_X

```
# Standardized data
Z = [[-1.22474487 -1.2493901 ]
 [-0.81649658 -0.85895569]
 [ 0.         -0.07808688]
 [ 0.40824829  0.70278193]
 [ 1.63299316  1.48365074]]
```

Effect of Standardization of Data



PCA: Implementation (contd)

- covariance matrix of the standardized data:
  Zt = np.transpose(Z)
  cov_matrix = np.cov(Zt)
  Σ = [[1.25 1.23530488]
  [1.23530488 1.25]]
- eigenvalues, eigenvectors = np.linalg.eig(cov_matrix)
  Eigenvalues: λ1 = 2.48530488, λ2 = 0.01469512
  Eigenvectors: v1 = [0.70710678, 0.70710678], v2 = [-0.70710678, 0.70710678]

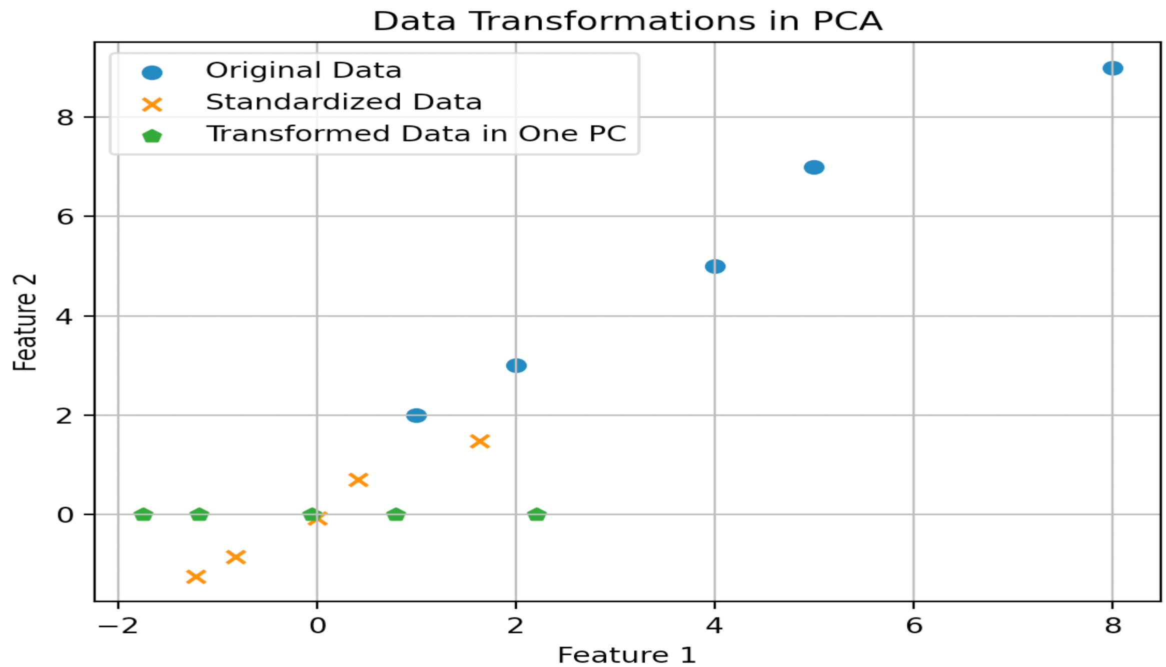
- sorted_indices = np.argsort(eigenvalues)[::-1]
- sorted_eigenvectors = eigenvectors[:, sorted_indices]
- Choose the top k (=1, in this example) eigenvectors as principal components:
  - principal_component = sorted_eigenvectors[:, 0] # First PC
    principal_component = [0.70710678, 0.70710678]

PCA: Implementation (contd)

- Project the original data onto the selected principal component:

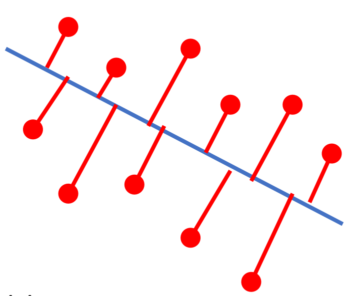
  - transformed_data = Z.dot(principal_component)
    [-1.74947761 -1.18472366 -0.05521576 0.785617 2.20380004]

- The transformed data now has a single dimension, representing the projection of the original data onto the principal component
- This lower-dimensional representation captures the most significant variance in the data.

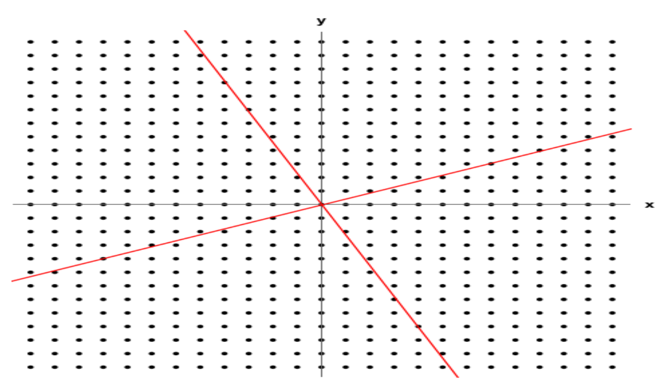


Relation to Linear Regression

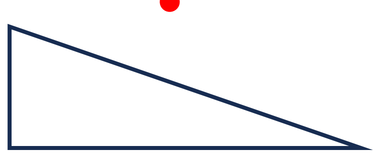
Linear Regression: Minimizing Least Squares



PCA: Maximizing the stretch of the projection

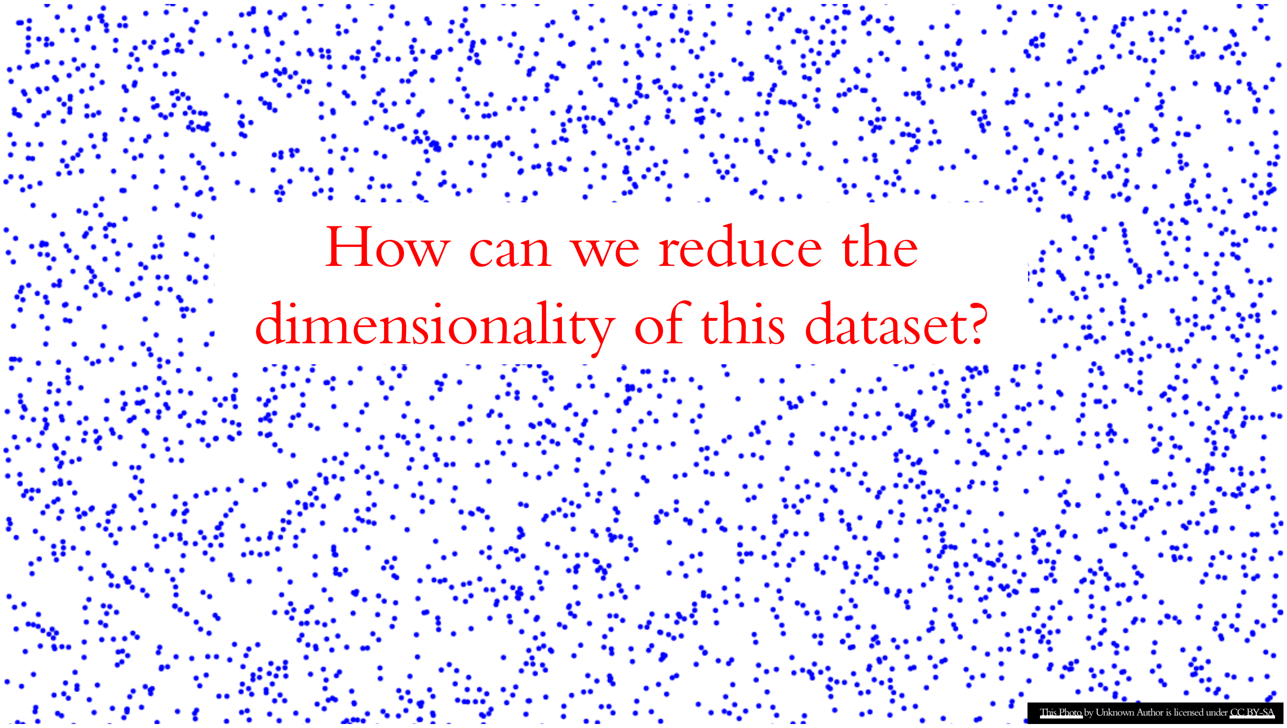


Minimizing this side (least square)



Is same as maximizing this side (projection)

Both are equivalent due to Pythagoras theorem

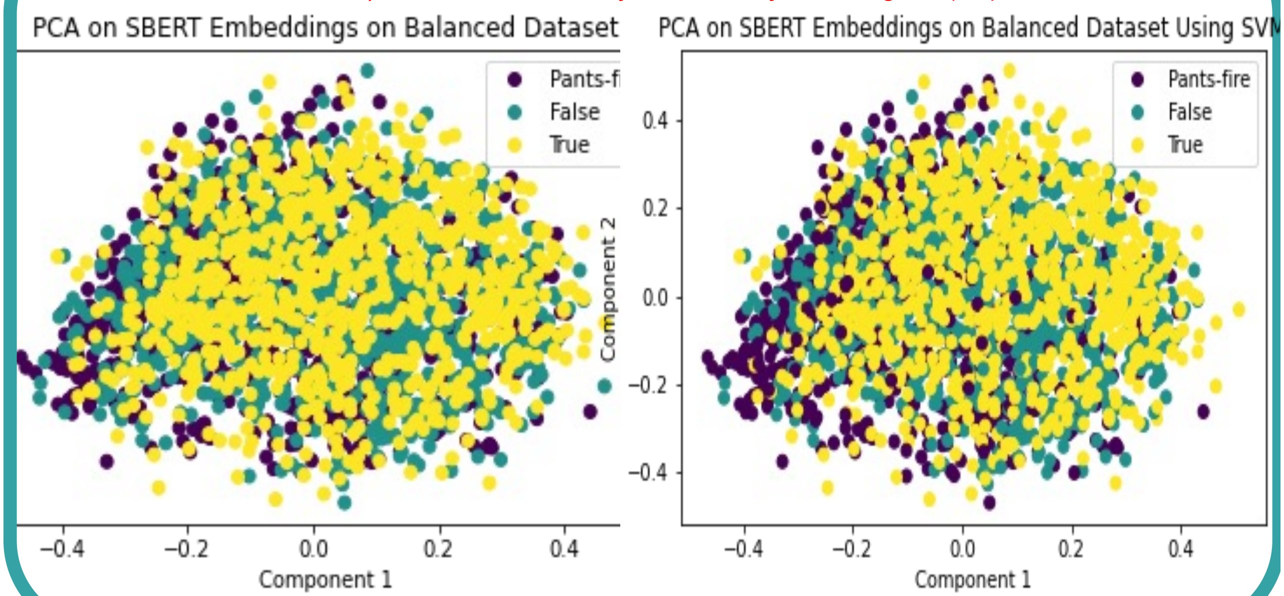


How can we reduce the dimensionality of this dataset?

Application of PCA

Pendyala, Vishnu S., and Foroozan Sadat Akhavan Tabatabaii. "Spectral analysis perspective of why misinformation containment is still an unsolved problem." *2023 IEEE Conference on Artificial Intelligence (CAI)*. IEEE, 2023.

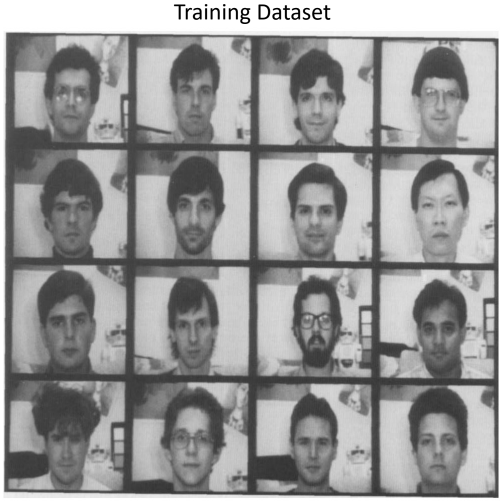
Pendyala, Vishnu S., and Foroozan Sadat Akhavan Tabatabaii. "Spectral analysis perspective of why misinformation containment is still an unsolved problem." 2023 IEEE Conference on Artificial Intelligence (CAI). IEEE, 2023.



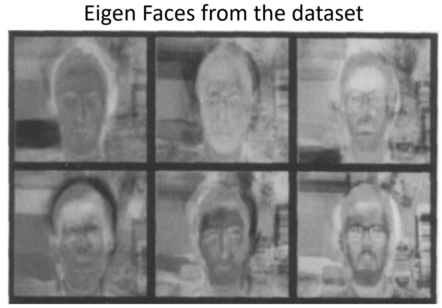
More Interesting Applications of PCA

Source: Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of cognitive neuroscience*, *3*(1), 71-86.

Training Dataset



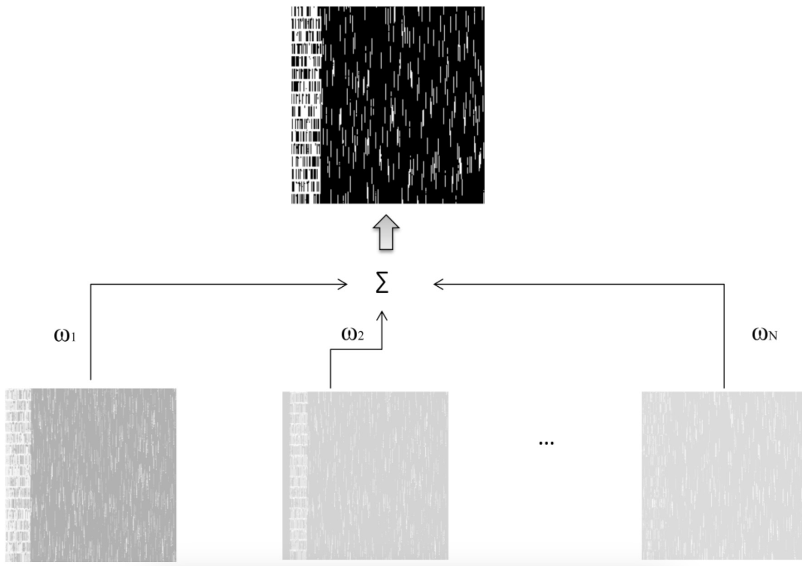
Eigen Faces from the dataset



Average Face

An original face image and its projection onto the face space defined by the eigenfaces

Source: Saleh, Mostafa E., A. Baith Mohamed, and A. Abdel Nabi. "Eigenviruses for metamorphic virus recognition." *IET information security* 5.4 (2011): 191-198.



- Eigenviruses are vectors that span across the most important features in the sample virus files
- Euclidean distance used to find the nearest neighbor for classification (recognition) of the virus

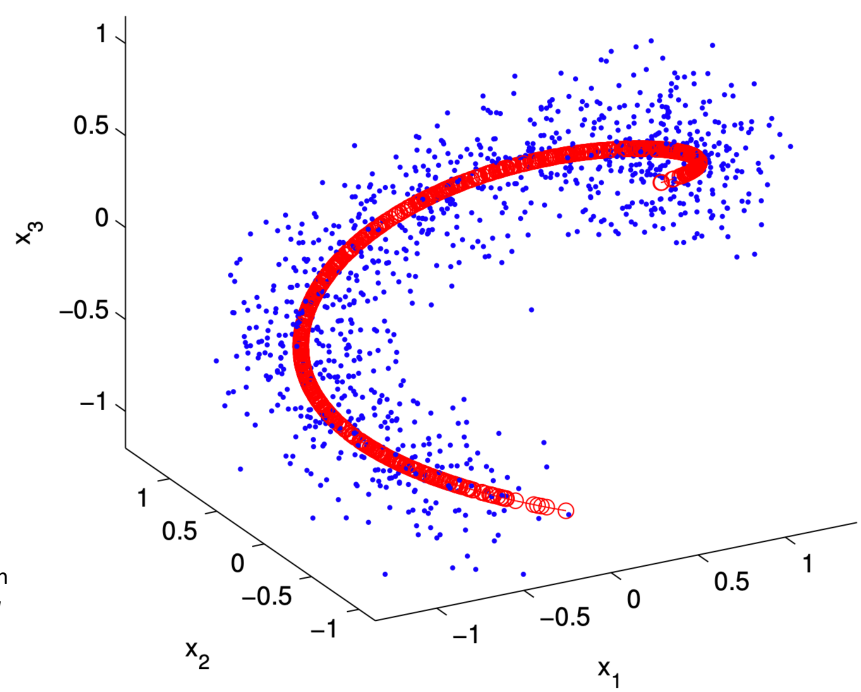
Other eigen* applications of PCA (contd)

- Speaker recognition and verification: Captures principal components of speech signals to create representative "**eigenvoices**."
  - Kuhn, Roland, et al. "Rapid speaker adaptation in eigenvoice space." *IEEE Transactions on Speech and Audio Processing* 8.6 (2000): 695-707.
  - Kwok, J., Mak, B., & Ho, S. (2003). Eigenvoice speaker adaptation via composite kernel principal component analysis. *Advances in Neural Information Processing Systems*, *16*.
- Hand gesture recognition: principal components from hand images or videos to generate "**eigenhands**."
  - Birk, Henrik, Thomas B. Moeslund, and Claus B. Madsen. "Real-time recognition of hand alphabet gestures using principal component analysis." Proceedings of the Scandinavian conference on image analysis. Vol. 1. Proceedings published by various publishers, 1997.

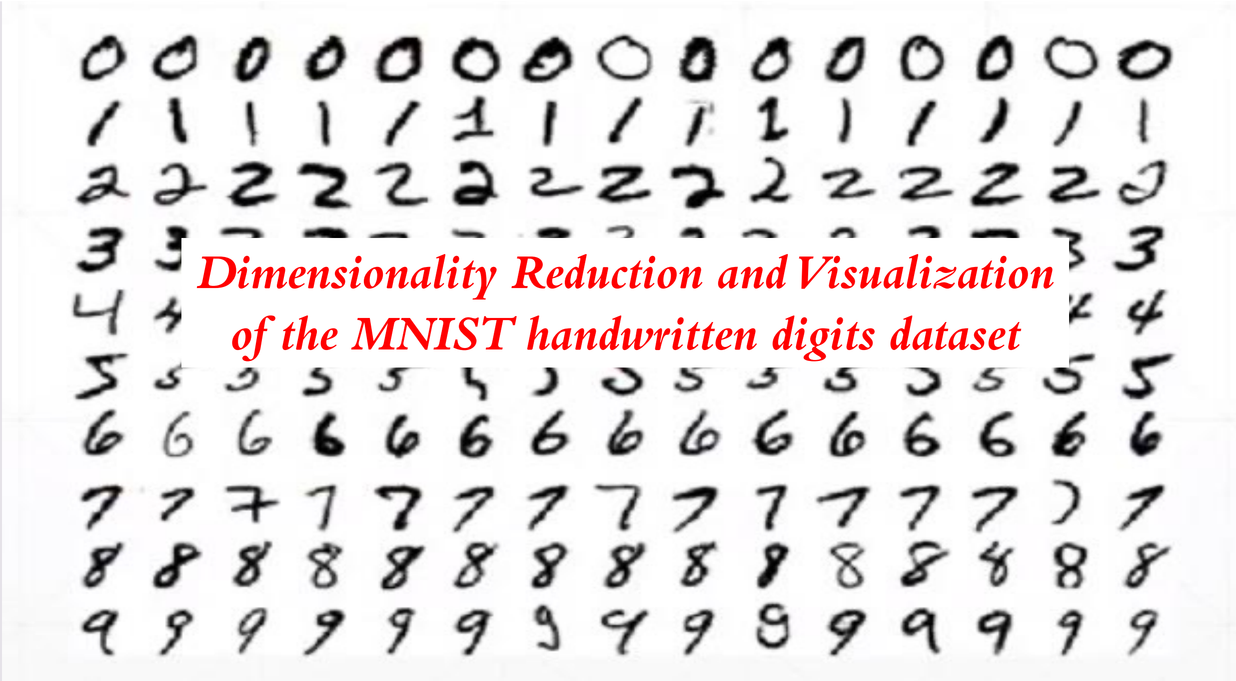
Other eigen* applications of PCA (contd)

- Gesture recognition in general: Captures principal directions of variation in gesture data to create "eigengestures."
  - Nakajima, Masato, et al. "Motion prediction based on eigen-gestures." Proc. of the 1st First Korea-Japan Joint Workshop on Pattern Recognition. 2006.
  - Gawron, Piotr, et al. "Eigengestures for natural human computer interface." Man-Machine Interactions 2. Springer Berlin Heidelberg, 2011.
- Texture analysis and synthesis: principal components of texture patterns to create "eigentextures."
  - Vasilescu, M. A. O., & Terzopoulos, D. (2004). TensorTextures: Multilinear image-based rendering. In ACM SIGGRAPH 2004 Papers (pp. 336-342).
- General image compression and representation: Computes eigenvectors of image covariance matrices to form a basis for representing images
  - Abadpour, A., & Kasaei, S. (2008). Color PCA eigenimages and their application to compression and watermarking. Image and Vision Computing, 26(7), 878-890.

Principal Components of Non-linear Data

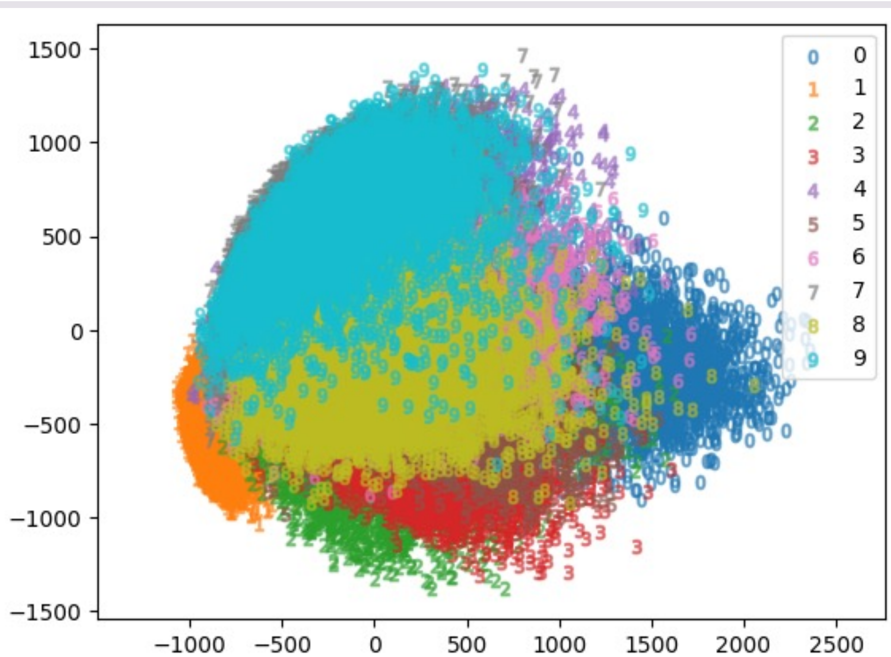


Source: Scholz, M. (2012). Validation of nonlinear PCA. *Neural processing letters*, *36*(1), 21-30.



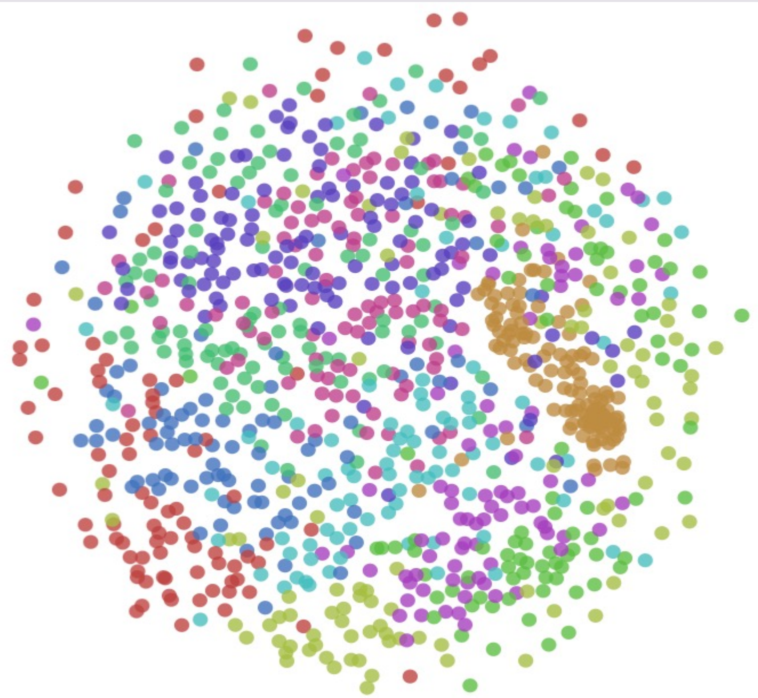
Dimensionality Reduction and Visualization of the MNIST handwritten digits dataset

PCA on ***labeled*** MNIST handwritten digits dataset visualized using the 1st and 2nd Principal Components



Multidimensional scaling (MDS) on ***labeled*** MNIST handwritten digits dataset visualized using the 1st and 2nd Principal Components

Source: https://colah.github.io/posts/2014-10-Visualizing-MNIST/

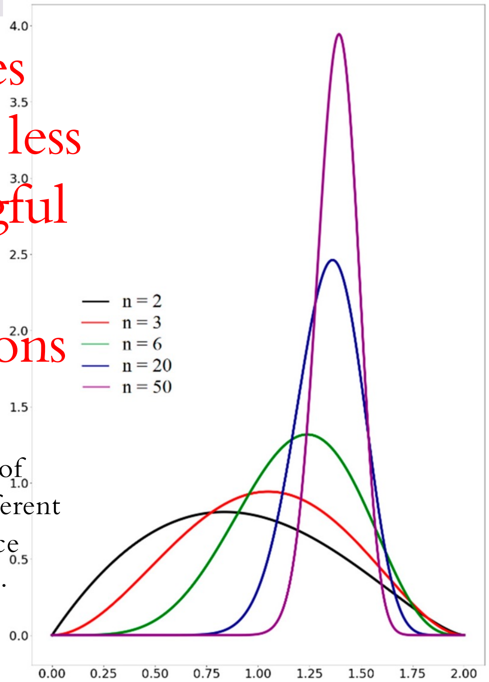


MDS is a spectral method that preserves the pair-wise distances between data items

What is the problem with the approach?

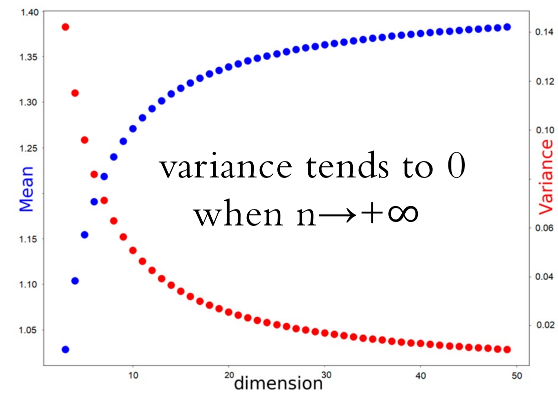
Distances become less meaningful in high dimensions

Distribution of distances for different values of space dimension n.



(a)

Lellouche, S., & Souris, M. (2019). Distribution of distances between elements in a compact set. *Stats*, *3*(1), 1-15.

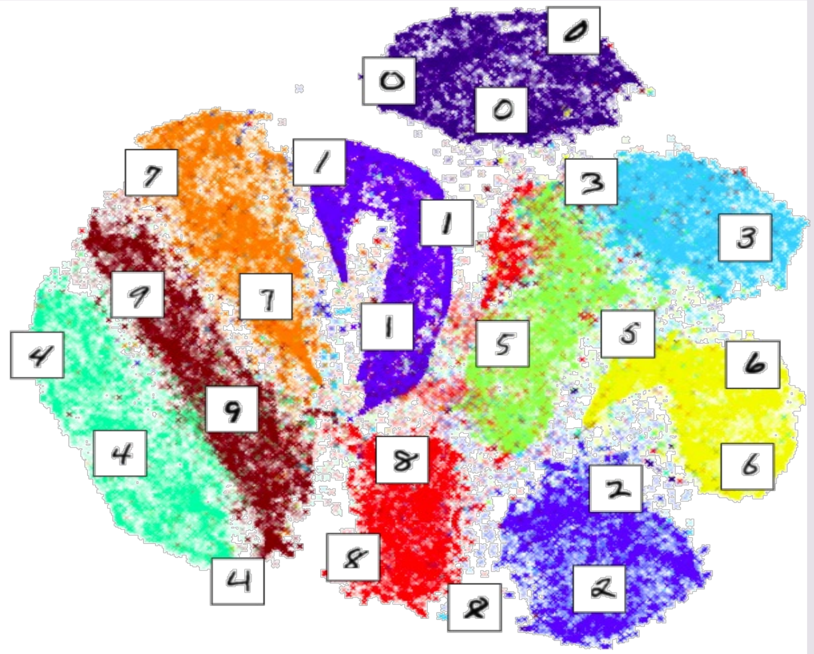


(b)

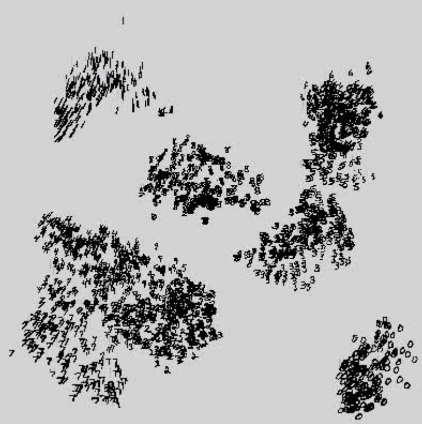
Solution: t-distributed stochastic neighbor embedding (t-SNE)

- Preserve only the smaller pairwise distances between data items (think neighbors within a group)
- Good at identifying clusters and anomalies, visualization of very high dimensional data
- Not good at preserving global distances like PCA does
- Handles non-linear relationships among data well
- Computation intensive
- Stochastic elements: initialization, stochastic gradient descent
- Highly configurable via hyperparameters and non-deterministic

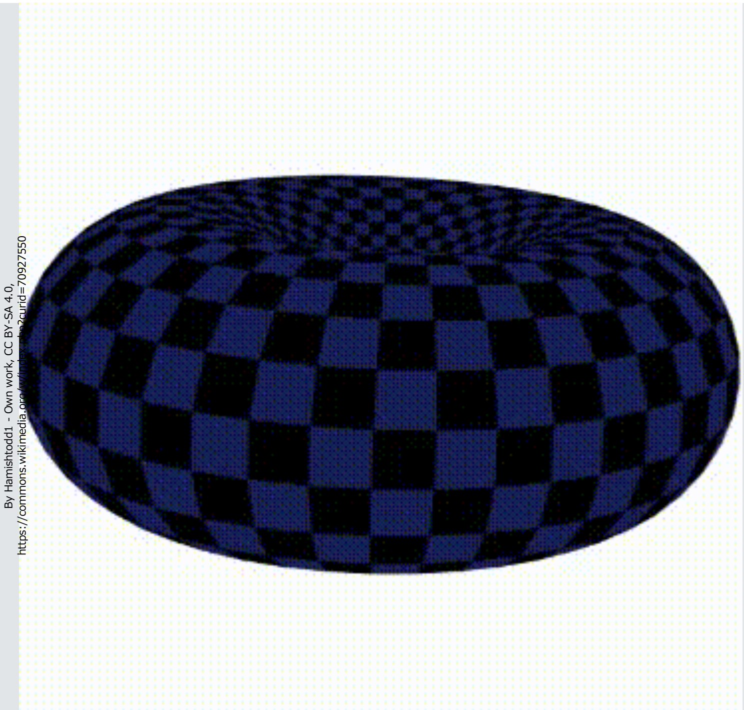
T-SNE on ***labeled*** MNIST handwritten digits dataset visualized using the 1st and 2nd Principal Components



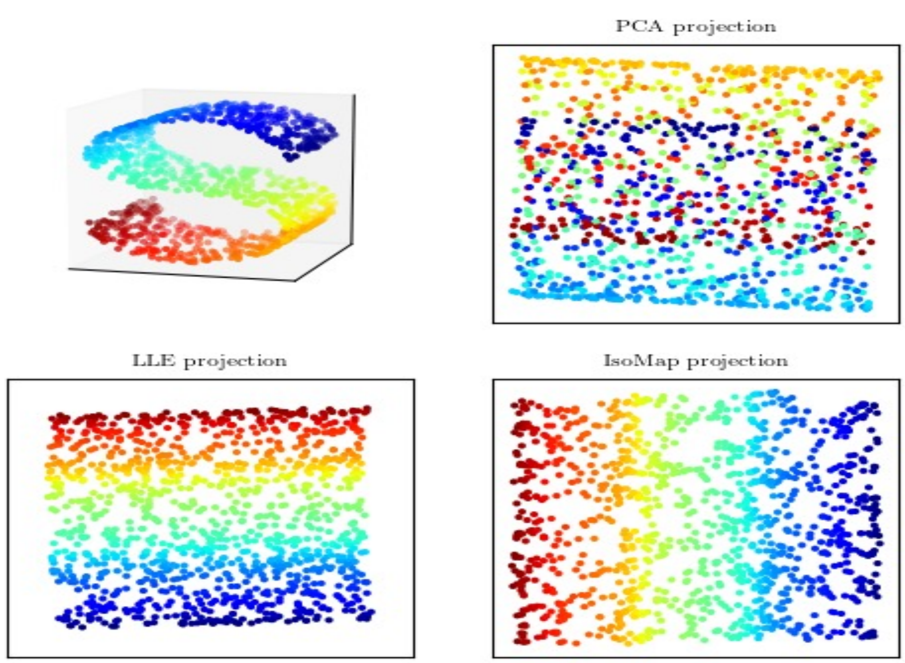
T-SNE on ***unlabeled*** MNIST handwritten digits dataset visualized using the 1st and 2nd Principal Components



Isomap uses Geodesic – the shortest distance between two points on a manifold surface, honoring the shape



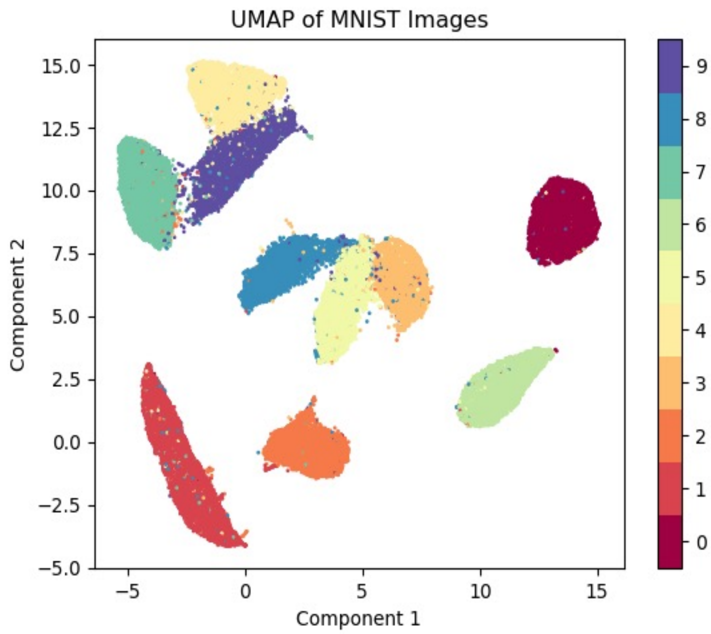
PCA vs Manifold Learning



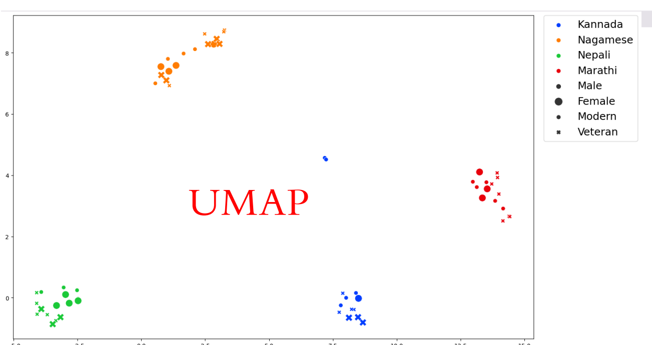
Source: https://www.astroml.org/_images/fig_S_manifold_PCA_1.png

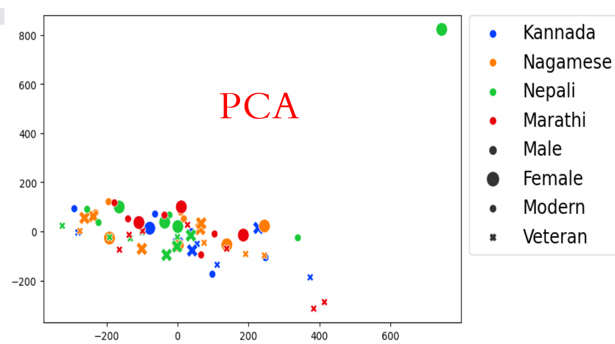
Uniform Manifold Approximation and Projection (UMAP)

- Builds a topological representation of data
- Preserves both local and global structure
- Based on manifold learning, graph theory
- Can capture non-linear relationships in the data
- Computationally efficient compared to t-SNE

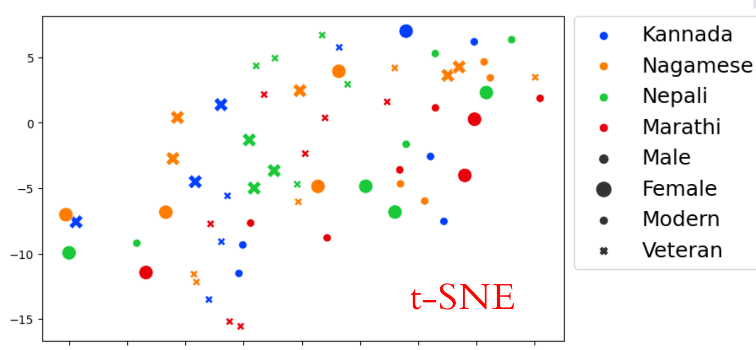


Source: Tezuka, Naoya, et al. "Resilience of Wireless Ad Hoc Federated Learning against Model Poisoning Attacks." *2022 IEEE 4th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*. IEEE, 2022.

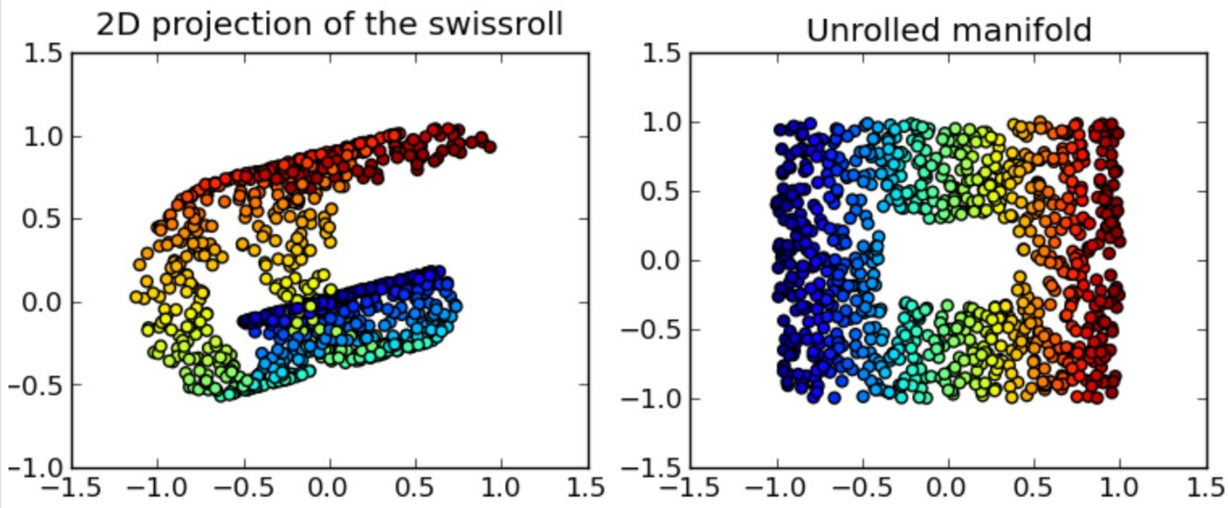




Source: Pendyala, Vishnu, Konduri Samhita, and Pendyala, Kriti, "Analysis of Multilanguage Regional Music Tracks using Representation Learning Techniques in Lower Dimensions." In *10th International Conference On Mathematics And Computing ICMC 2024. Springer. [Best paper award]*



Many algorithms for non-linear dimensionality reduction

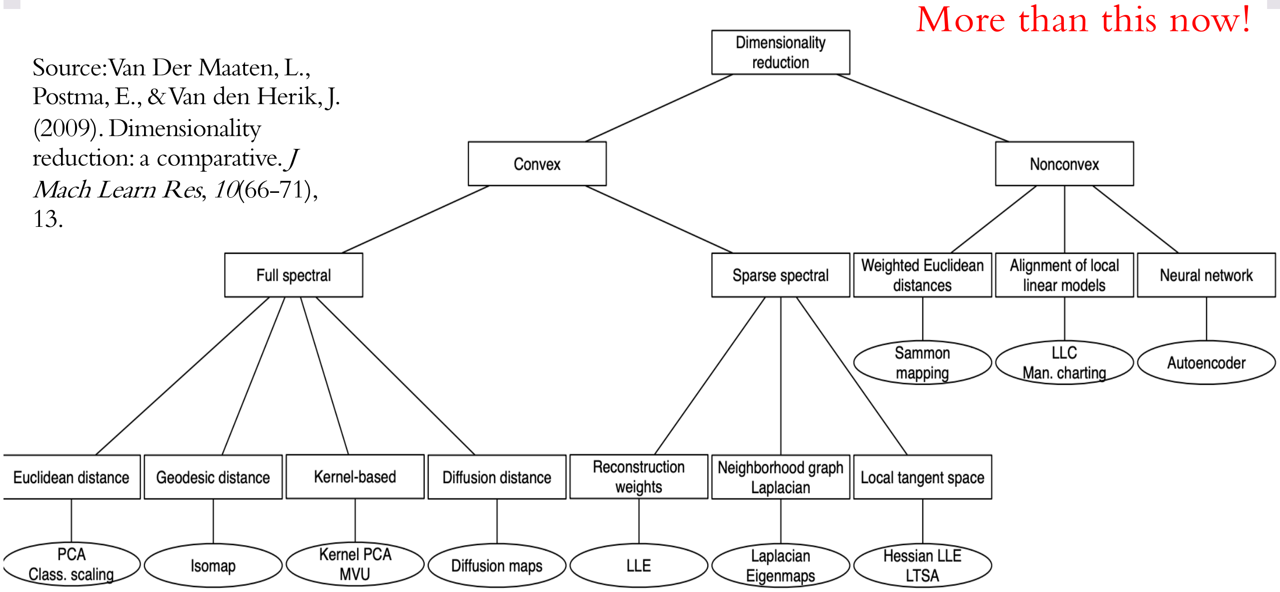


Source: https://en.wikipedia.org/wiki/Nonlinear_dimensionality_reduction#/media/File:Lle_hlle_swissroll.pngCC BY 3.0

Taxonomy of dimensionality reduction techniques (2009)

More than this now!

Source:Van Der Maaten, L., Postma, E., & Van den Herik, J. (2009). Dimensionality reduction: a comparative. *J Mach Learn Res*, *10*(66-71), 13.



Kernel PCA

Increase the dimensions: Transform the non-linear data into higher dimensional space

Reduce the dimensions: Run PCA in higher dimensional space

Finding the eigen values of the kernel matrix is equivalent to finding those of the covariance matrix

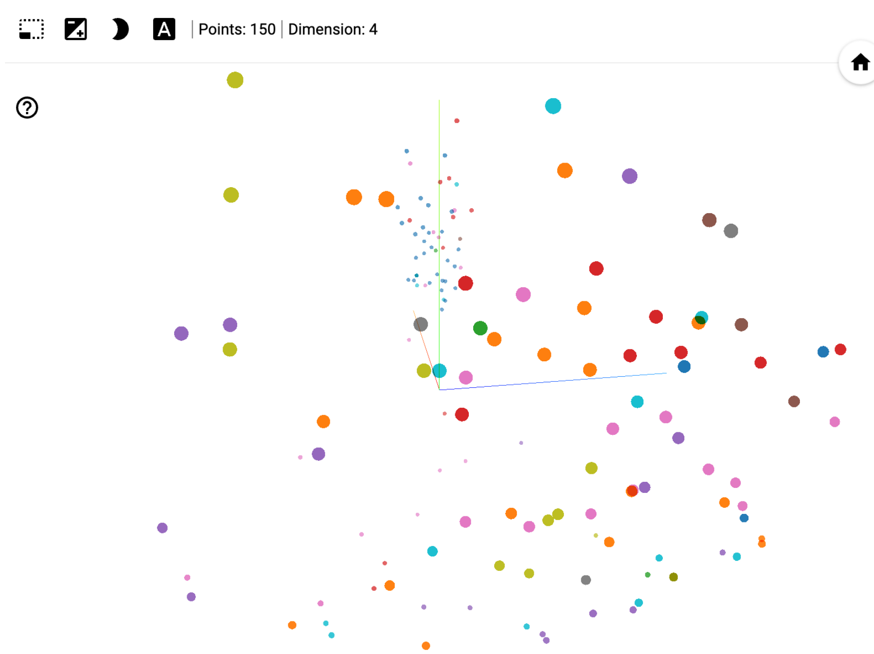
Use the kernel matrix in place of covariance matrix and compute the principal components in the transformed space

The feature vector x is transformed as:

$$\sum_{i=1}^{N} \alpha_i K(x, x^{(i)}).$$

α_i are the respective weights (values in the eigen vectors of the kernel matrix)

Demo time: Tensorflow Embedding Projector




SAN JOSE
STATE
UNIVERSITY
https://www.sjsu.edu/people/vishnu.pendyala/
@vishnupendyala
This Photo by Unknown Author is licensed under CC BY-SA