ECEN 758 Data Mining and Analysis: Lecture 4, Dimensionality Reduction

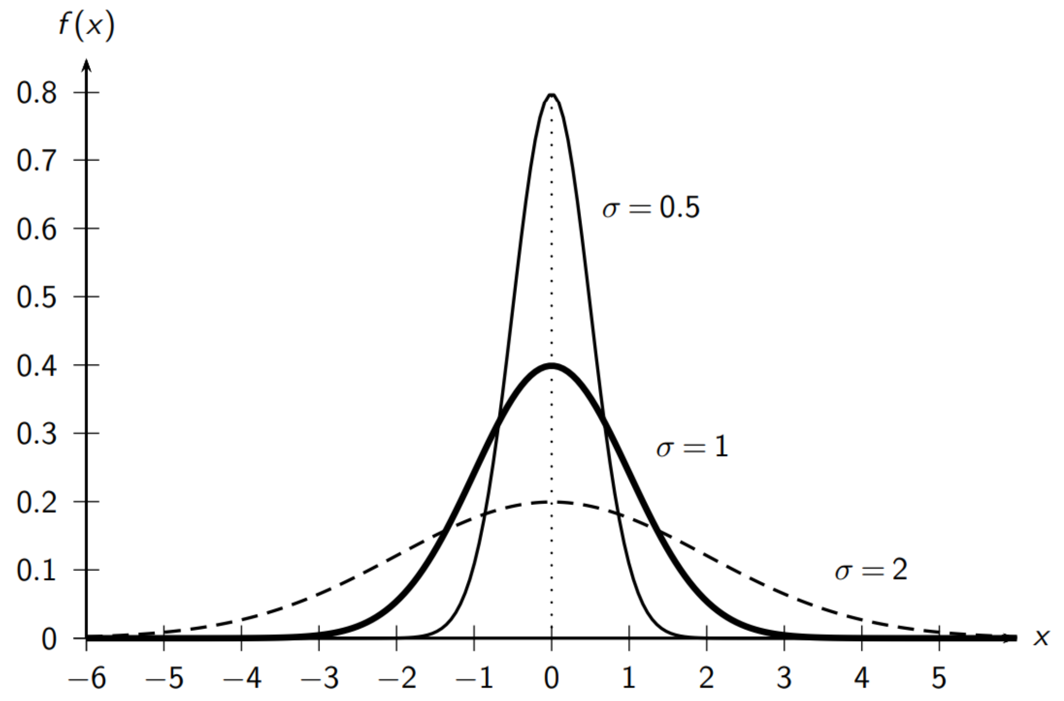
Joshua Peeples, Ph.D.
Assistant Professor
Department of Electrical and Computer Engineering

Announcements

- Assignment #1 is available today (08/28)
  - Due next Friday, 09/06
- Submit PDF for solutions (can include code in PDF or submit as separate file)

Last Lecture

- Numerical attributes
  - Normal distribution
- Categorical attributes



Today

TEXAS A&M UNIVERSITY Engineering

- Dimensionality reduction
- Reading: ZM Chapter 7
- Supplemental reading: ZM Chapter 6

What are features?

Feature Extraction

Learning from experience

0: Macaw 1: Conure

Feature Extraction

TEXAS A&M UNIVERSITY Engineering

- Ideally, as you add more features, the data becomes more separable
- We’ll explore this more for support vector machines

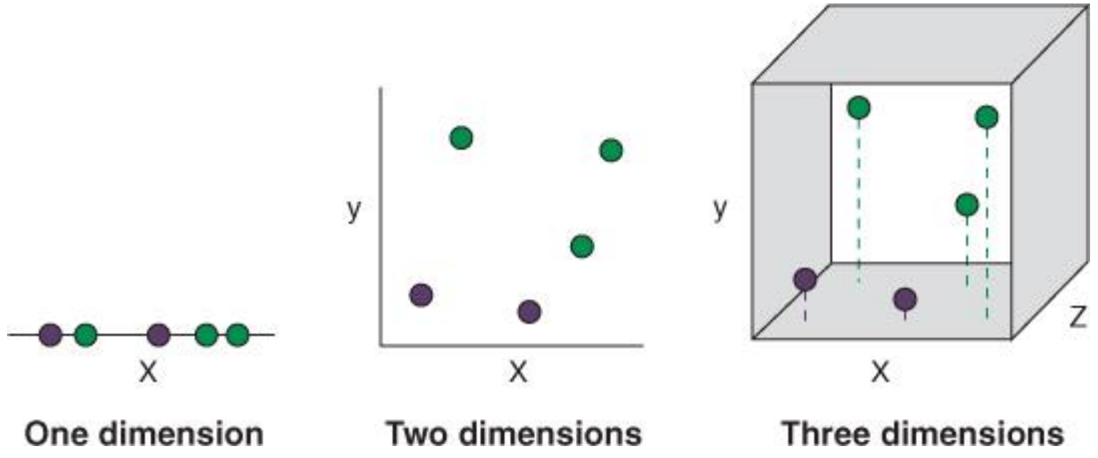


Image from: H. Rhys, Machine Learning with R, the tidyverse and mlr

Curse of Dimensionality

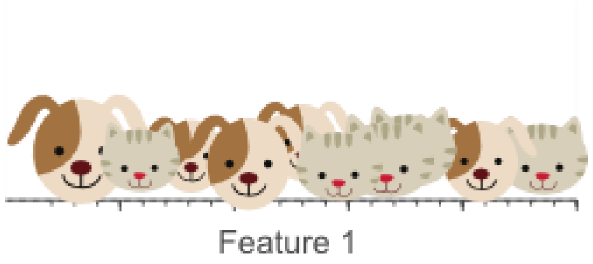
Cats vs Dogs

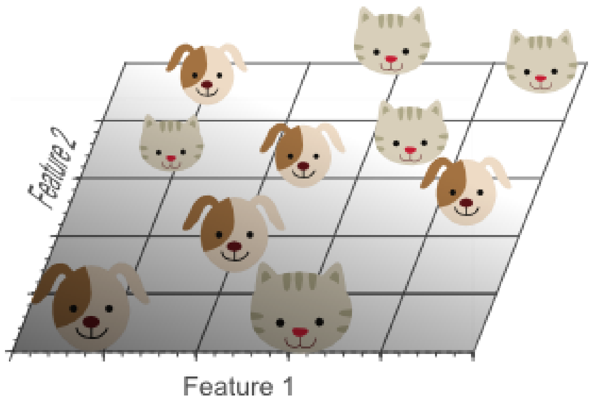
- What features would you use to distinguish between cats and dogs?

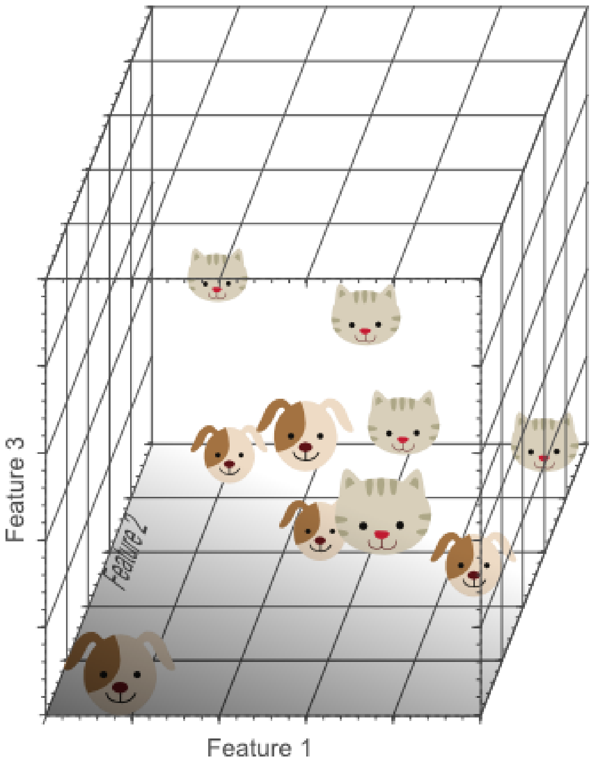
Image from: Cats vs. Dogs: Which One Is Right For You.

Increasing Features

- To improve performance, we can increase the number of features



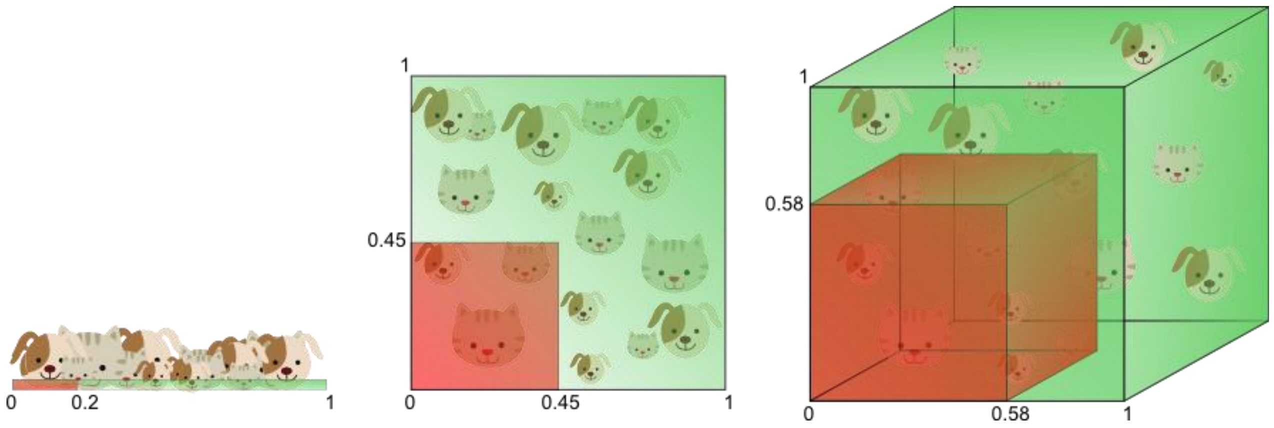




Images from: The Curse of Dimensionality in Classification.

Curse of Dimensionality

- As we increase the number of features, we need more data
- Grow exponentially as the feature dimension increases



Images from: The Curse of Dimensionality in Classification.

Performance Saturation

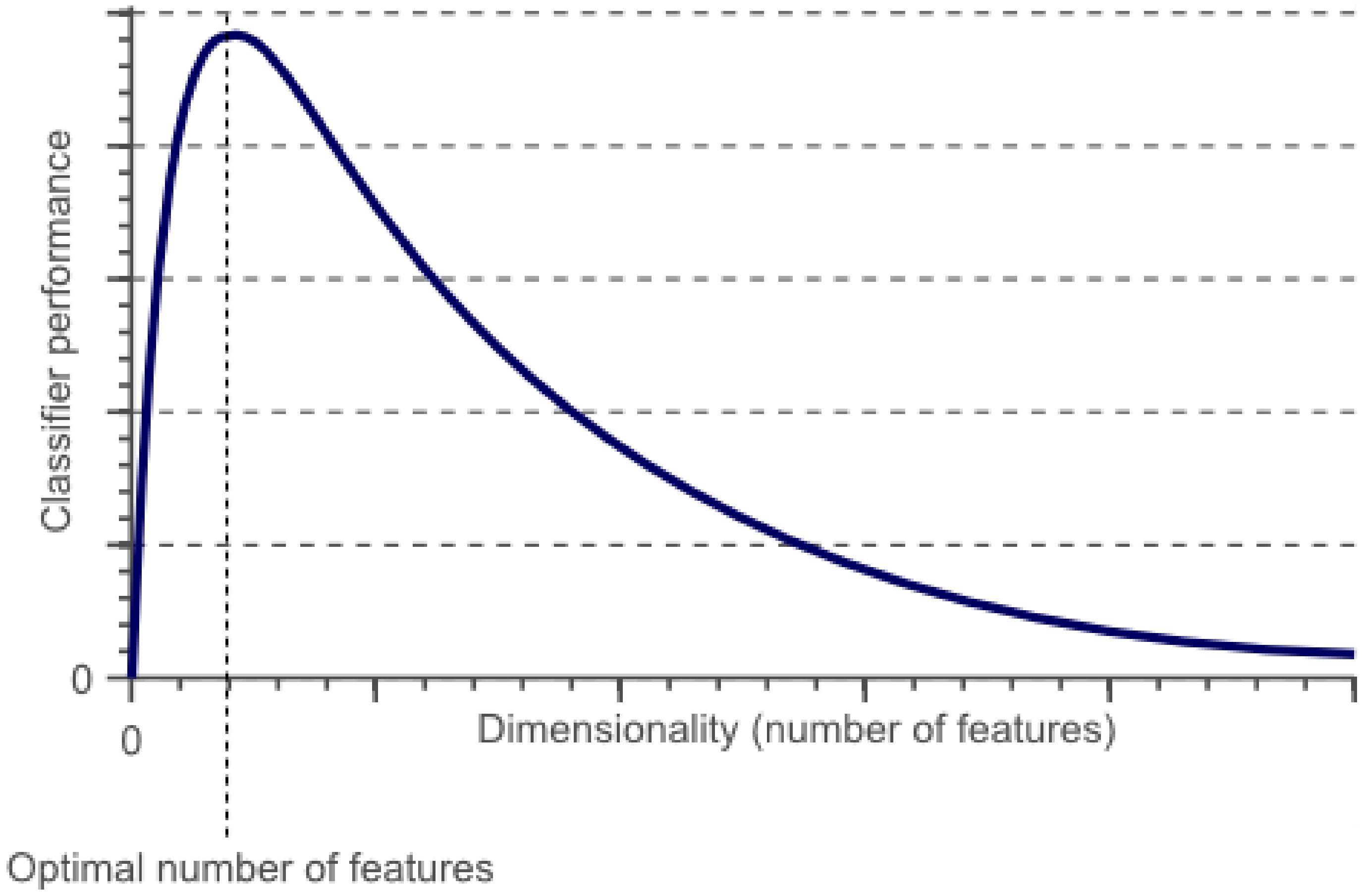
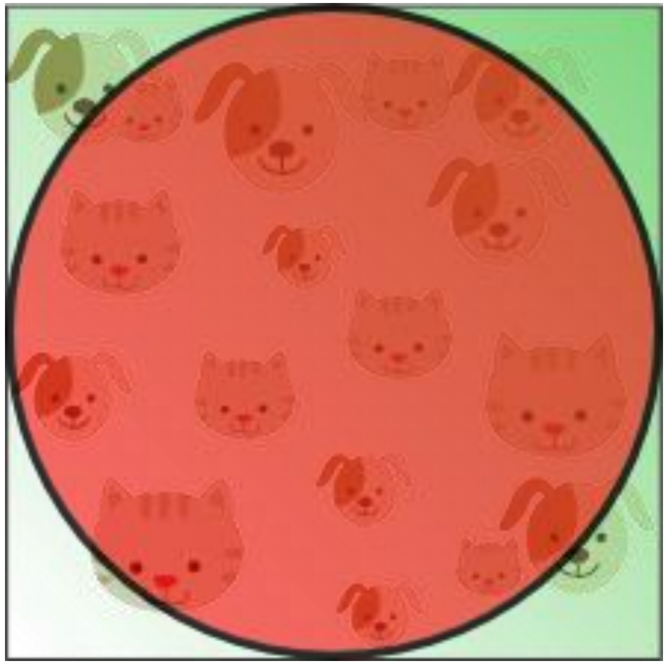


Image from: The Curse of Dimensionality in Classification.

Curse of Dimensionality

- Feature space lies on unit square (2D)
- Average of feature space is the center of square
- Samples not in unit circle are harder to classify



Images from: The Curse of Dimensionality in Classification.

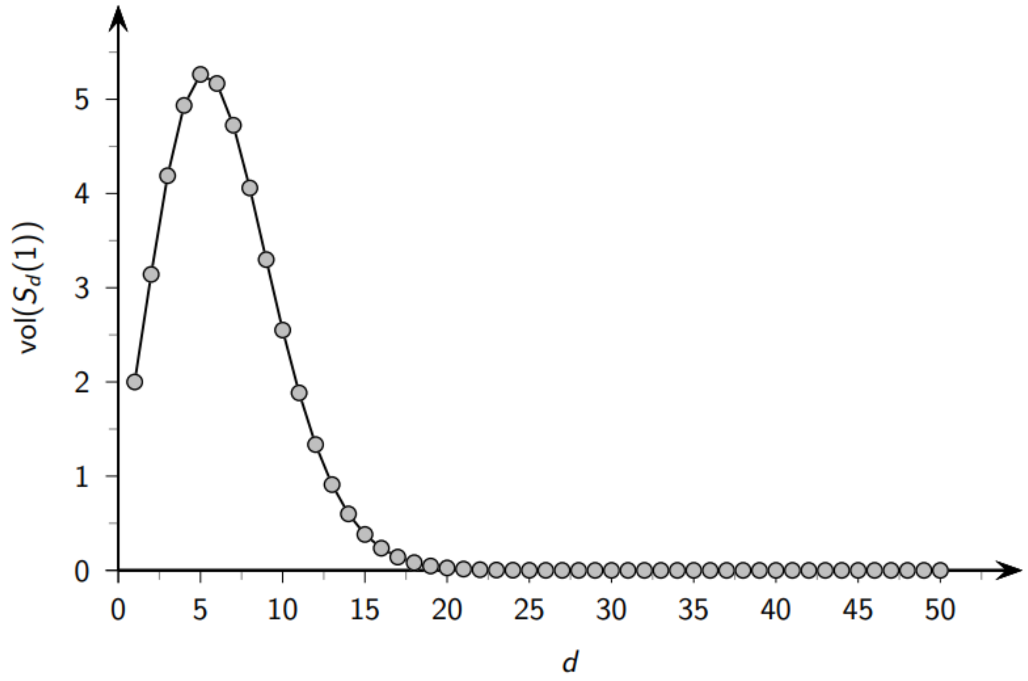
Curse of Dimensionality

TEXAS A&M UNIVERSITY Engineering

- The volume of the circle (hypersphere) with the feature dimension (for radius 1)

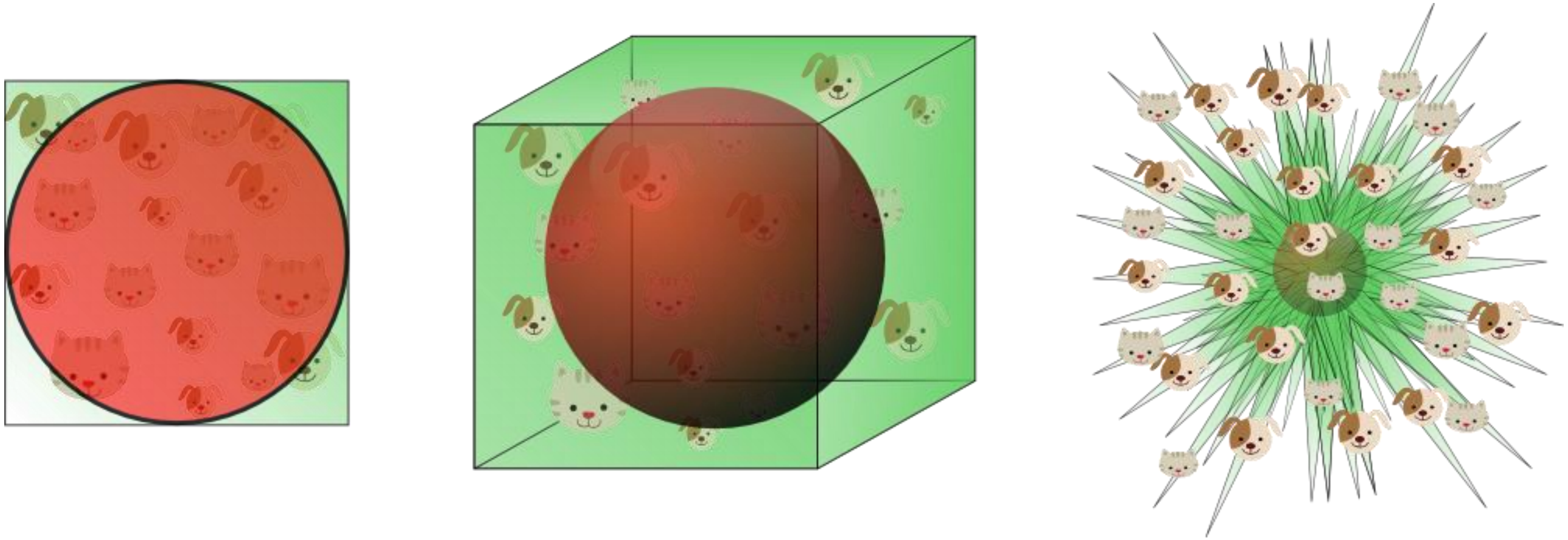
$$\mathrm{vol}(S_d(r)) = K_d r^d = \left(\frac{\pi^{\frac{d}{2}}}{\Gamma\left(\frac{d}{2}+1\right)}\right) r^d$$

$$\Gamma\left(\frac{d}{2}+1\right) = \begin{cases} \left(\frac{d}{2}\right)! & \text{if } d \text{ is even} \\ \sqrt{\pi}\left(\frac{d!!}{2^{(d+1)/2}}\right) & \text{if } d \text{ is odd} \end{cases}$$



Curse of Dimensionality

- Volume of hypersphere goes to zero as dimensionality increases

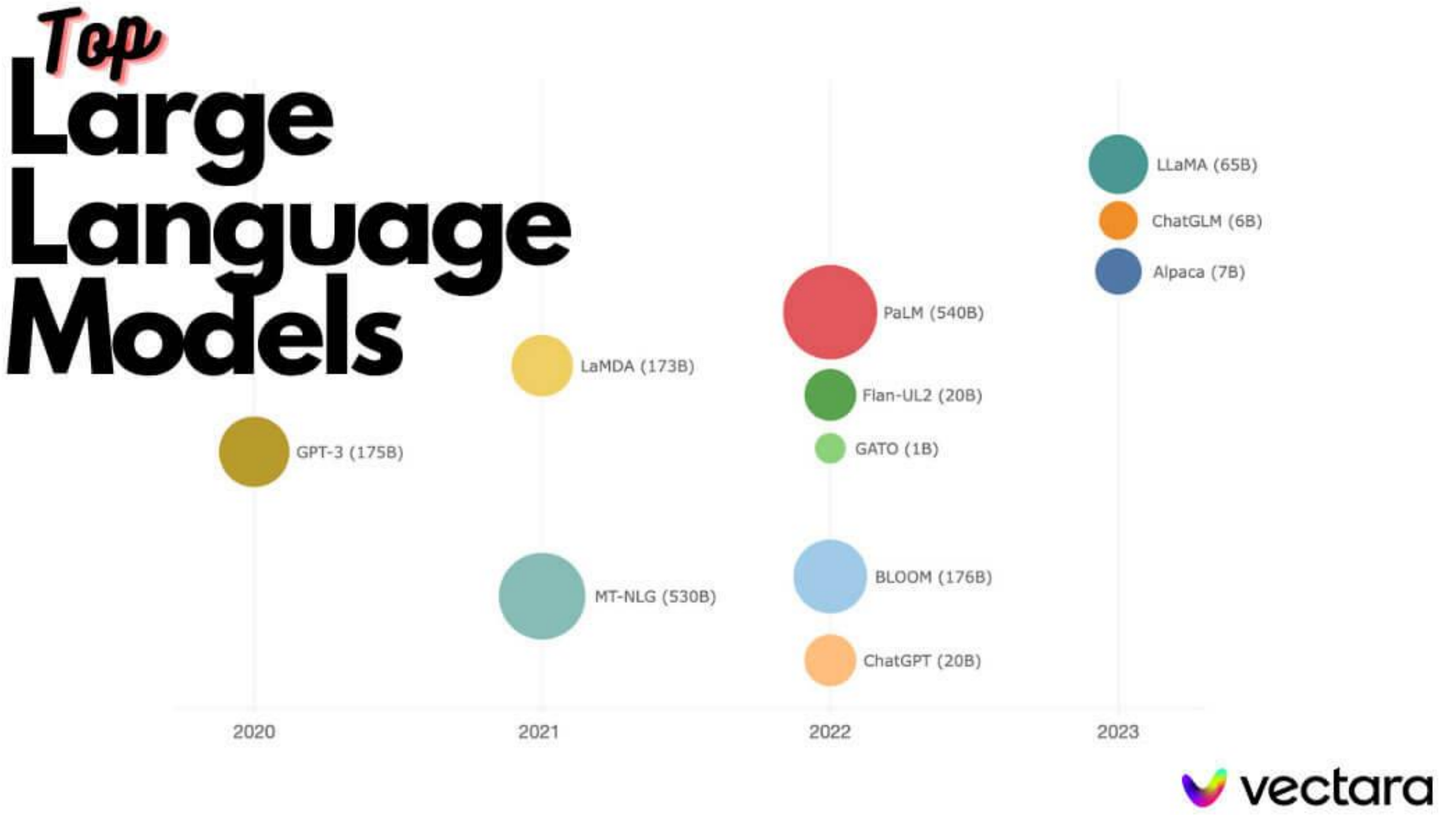


2D 3D 8D

Images from: The Curse of Dimensionality in Classification.

Current State-of-the-Art Models

- Deep learning approaches have many parameters (e.g., features)
- “Data-hungry” approaches



How can we mitigate Curse of Dimensionality?

Dimensionality Reduction

TEXAS A&M UNIVERSITY Engineering

- Goal: find lower dimensional representation of data matrix **D**
- Can we express data using set of orthonormal vectors, **U**
- **a** represents data **x** in new basis

$$\boldsymbol{x} = a_1 \boldsymbol{u}_1 + a_2 \boldsymbol{u}_2 + \cdots + a_d \boldsymbol{u}_d$$

$$\boldsymbol{x} = \boldsymbol{U}\boldsymbol{a}$$

Image from: QuestionPro

Dimensionality Reduction

- Infinite choices for orthonormal basis
- Goal is to find optimal basis that preserves most important information of data
- New dimension *r* should be less than *d*
- ***P*** is the projection matrix

$$\mathbf{x}' = a_1\mathbf{u}_1 + a_2\mathbf{u}_2 + \cdots + a_r\mathbf{u}_r = \sum_{i=1}^{r} a_i\mathbf{u}_i = \mathbf{U}_r\mathbf{a}_r$$

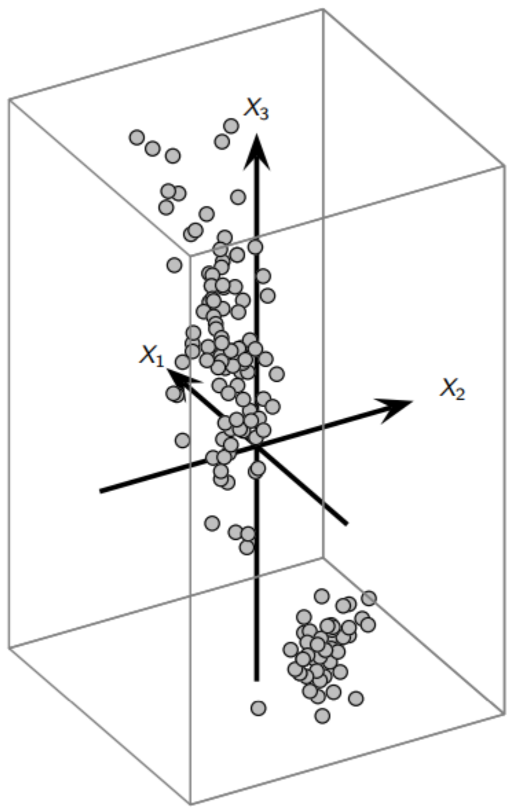
$$\boxed{\mathbf{x}' = \mathbf{U}_r\mathbf{U}_r^T\mathbf{x} = \mathbf{P}_r\mathbf{x}}$$

Dimensionality Reduction: Error Vector

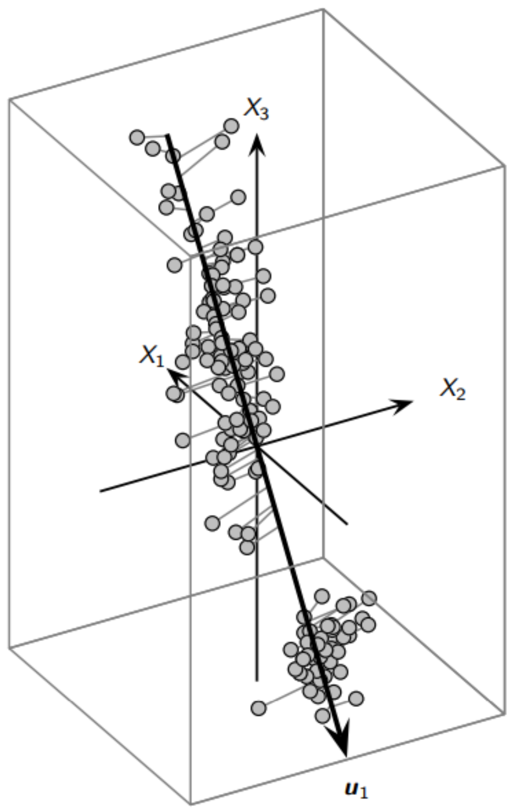
- Find projection that minimizes error vector

$$\epsilon = \sum_{i=r+1}^{d} a_i \boldsymbol{u}_i = \mathbf{x} - \mathbf{x}'$$

Iris Data: Optimal 1D Basis



Iris Data: 3D



Optimal 1D Basis

Principal Component Analysis

Principal Component Analysis (PCA)

- Seek projection that best captures variance
- Direction with the largest projected variance is first principal component
- Direction that maximizes variance should minimize error

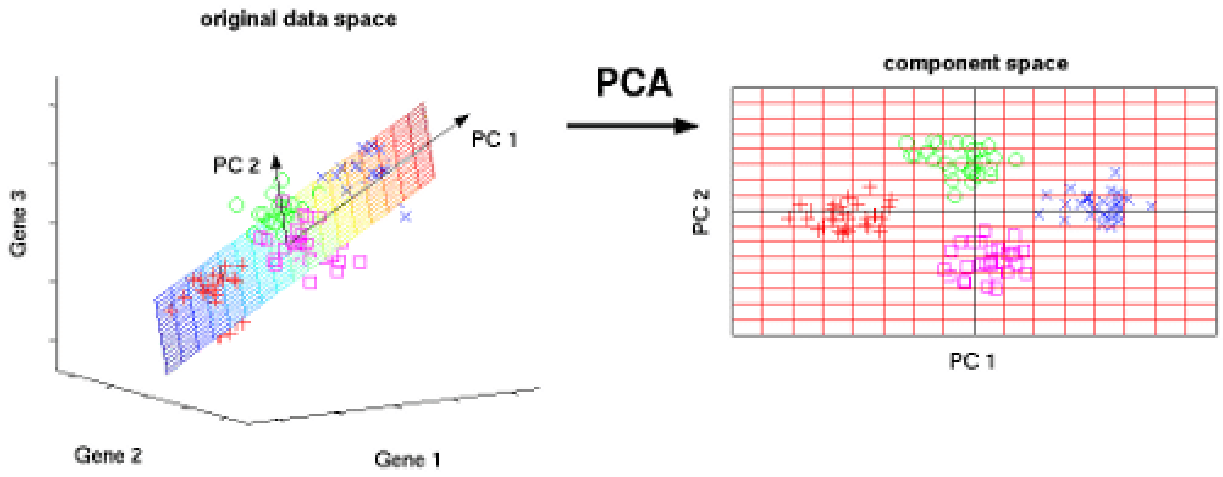


Image from: M. Scholz, Approaches to analyse and interpret biological profile data.

Principal Component Analysis (PCA)

- Find unit vector **u** that maximizes projected variance
- Data need to first be centered
- Computed projected variance along **u**

$$\sigma_{\boldsymbol{u}}^2 = \frac{1}{n}\sum_{i=1}^{n}(a_i - \mu_{\boldsymbol{u}})^2 = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{u}^T\left(\boldsymbol{x}_i\boldsymbol{x}_i^T\right)\boldsymbol{u} = \boldsymbol{u}^T\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^T\right)\boldsymbol{u} = \boldsymbol{u}^T\Sigma\boldsymbol{u}$$

Principal Component Analysis (PCA)

- Maximize projected variance J(**u**)
- Constraint of $\boldsymbol{u}^T\boldsymbol{u} = 1$

$$\max_{\boldsymbol{u}} J(\boldsymbol{u}) = \boldsymbol{u}^T \Sigma \boldsymbol{u} - \alpha(\boldsymbol{u}^T\boldsymbol{u} - 1)$$

Optimizing Objective Function

- In machine learning, typically want to minimize or maximize objective function
- Achieved by setting the derivative of objective function with variable of interest

$$\frac{\partial}{\partial \boldsymbol{u}}\left(\boldsymbol{u}^T \Sigma \boldsymbol{u} - \alpha(\boldsymbol{u}^T \boldsymbol{u} - 1)\right) = 0$$

that is, $2\Sigma \boldsymbol{u} - 2\alpha \boldsymbol{u} = 0$

which implies $\boxed{\Sigma \boldsymbol{u} = \alpha \boldsymbol{u}}$

PCA: Direction of Most Variance

- Maximizing the projected variance means:
  - Selecting largest eigenvalue of covariance matrix
  - Dominant eigenvector is the direction of most variance (first principal component)

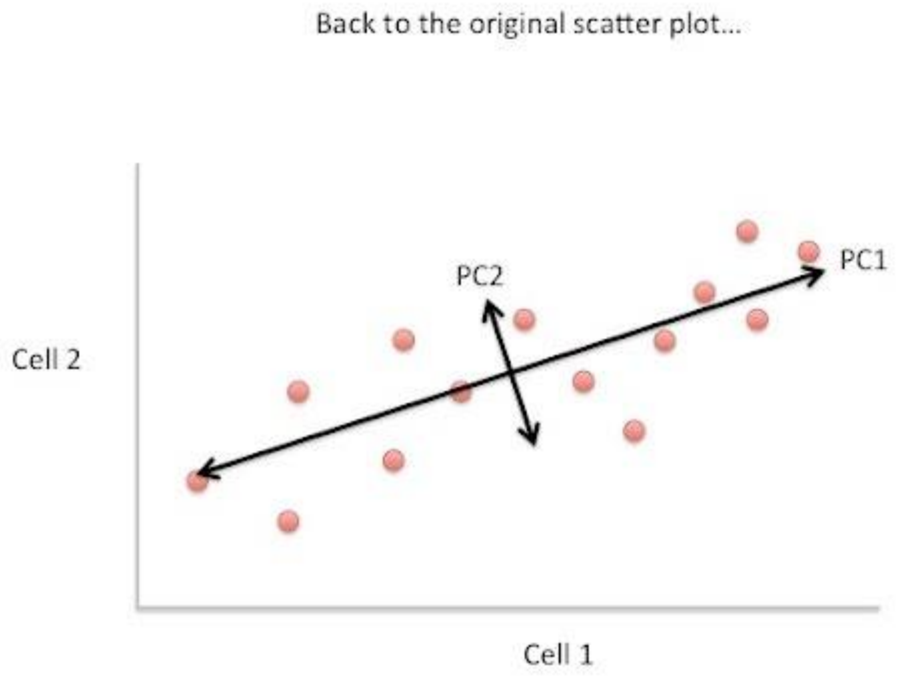


Image from: J. Starmer, PCA Clearly Explained.

PCA: Minimum Squared Error

PCA: Minimum Square Error Approach

- Maximizing the variance also minimizes the squared error

$$MSE(\boldsymbol{u}) = \frac{1}{n}\sum_{i=1}^{n} ||\epsilon_i||^2 = \frac{1}{n}\sum_{i=1}^{n} ||\boldsymbol{x}_i - \boldsymbol{x}_i'||^2 = \sum_{i=1}^{n} \frac{||\boldsymbol{x}_i||^2}{n} - \boldsymbol{u}^T \Sigma \boldsymbol{u}$$

PCA: Minimum Square Error Approach

- First term is fixed for **D**
- Same solution for maximization of variance and minimization of squared error

$$\sum_{i=1}^{n} \frac{\|\mathbf{x}_i\|^2}{n} - \mathbf{u}^T \Sigma \mathbf{u} = var(\mathbf{D}) = tr(\Sigma) = \sum_{i=1}^{d} \sigma_i^2$$

$$MSE(\mathbf{u}_1) = var(\mathbf{D}) - \mathbf{u}_1^T \Sigma \mathbf{u}_1 = var(\mathbf{D}) - \lambda_1$$

PCA: 2-D Projection

- 2D subspace captures the most variance in **D** with the two eigenvectors that correspond to largest and second largest eigenvalues

$$\boldsymbol{a}_i = \boldsymbol{U}_2^T \boldsymbol{x}_i$$

$$var(\boldsymbol{A}) = \boldsymbol{u}_1^T \Sigma \boldsymbol{u}_1 + \boldsymbol{u}_2^T \Sigma \boldsymbol{u}_2 = \boldsymbol{u}_1^T \lambda_1 \boldsymbol{u}_1 + \boldsymbol{u}_2^T \lambda_2 \boldsymbol{u}_2 = \lambda_1 + \lambda_2$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{x}_i'\|^2 = var(\boldsymbol{D}) - \frac{1}{n}\sum_{i=1}^{n} (\boldsymbol{x}_i^T \boldsymbol{P}_2 \boldsymbol{x}_i) = var(\boldsymbol{D}) - var(\boldsymbol{A})$$

PCA: r-D Projection

- r-D subspace captures the most variance in **D** with the r eigenvectors that correspond to r-largest eigenvalues

$$var(\boldsymbol{A}) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i^T \boldsymbol{P}_r \boldsymbol{x}_i = \sum_{i=1}^{r} \boldsymbol{u}_i^T \Sigma \boldsymbol{u}_i = \sum_{i=1}^{r} \lambda_i$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \left\|\boldsymbol{x}_i - \boldsymbol{x}_i'\right\|^2 = var(\boldsymbol{D}) - \sum_{i=1}^{r} \lambda_i = \sum_{i=1}^{d} \lambda_i - \sum_{i=1}^{r} \lambda_i$$

PCA Algorithmic Design

Choosing the Dimensionality

- To select the appropriate dimension, use ratio of total variance captured by the first r-components
- If you want to capture 90% of the variance in your data, ratio should be at least 0.9

$$f(r) = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_r}{\lambda_1 + \lambda_2 + \cdots + \lambda_d} = \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{d} \lambda_i} = \frac{\sum_{i=1}^{r} \lambda_i}{var(\mathbf{D})}$$

PCA Algorithm

PCA (D, α)**:**

$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$ `// compute mean`

$\mathbf{Z} = \mathbf{D} - \mathbf{1} \cdot \boldsymbol{\mu}^T$ `// center the data`

$\Sigma = \frac{1}{n}\left(\mathbf{Z}^T\mathbf{Z}\right)$ `// compute covariance matrix`

$(\lambda_1, \lambda_2, \ldots, \lambda_d) = \text{eigenvalues}(\Sigma)$ `// compute eigenvalues`

$\mathbf{U} = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_d \end{pmatrix} = \text{eigenvectors}(\Sigma)$ `// compute eigenvectors`

$f(r) = \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$, for all $r = 1, 2, \ldots, d$ `// fraction of total variance`

Choose smallest r so that $f(r) \geq \alpha$ `// choose dimensionality`

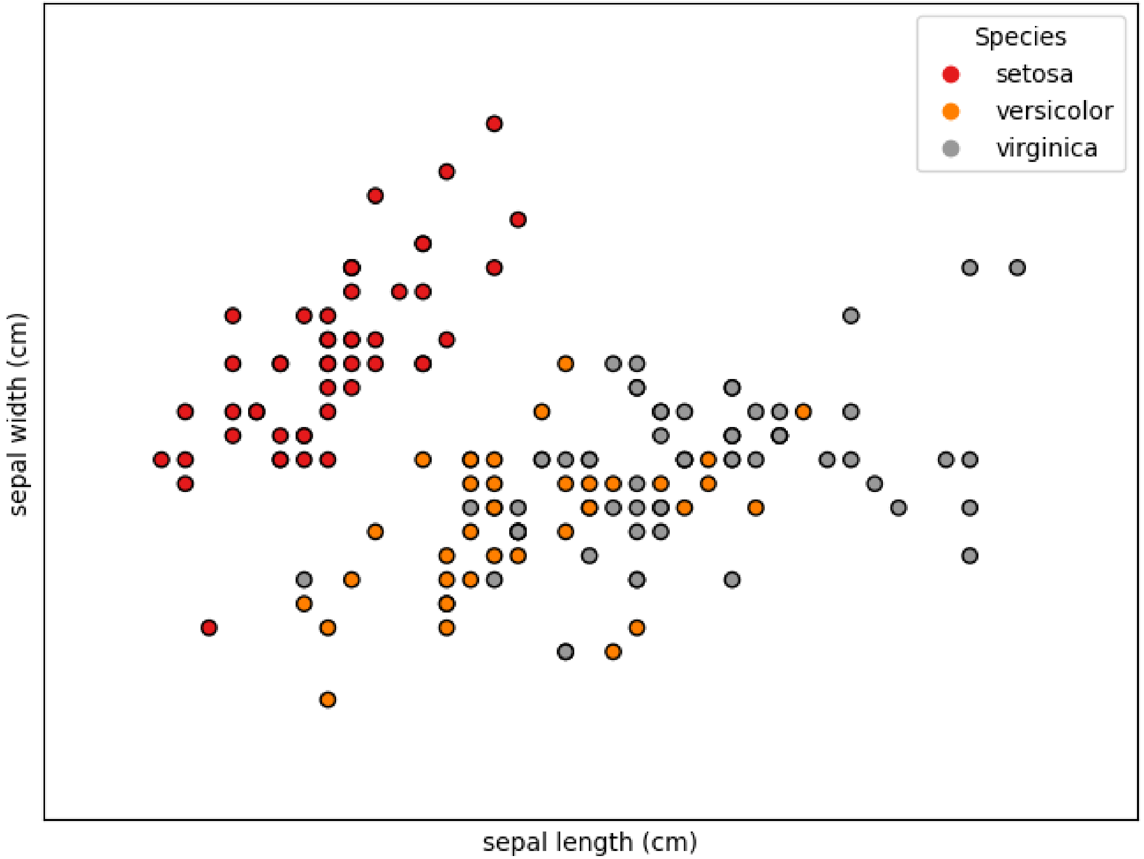
$\mathbf{U}_r = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_r \end{pmatrix}$ `// reduced basis`

$\mathbf{A} = \{\mathbf{a}_i \mid \mathbf{a}_i = \mathbf{U}_r^T \mathbf{x}_i, \text{for } i = 1, \ldots, n\}$ `// reduced dimensionality data`

PCA Example

Iris Flower Dataset

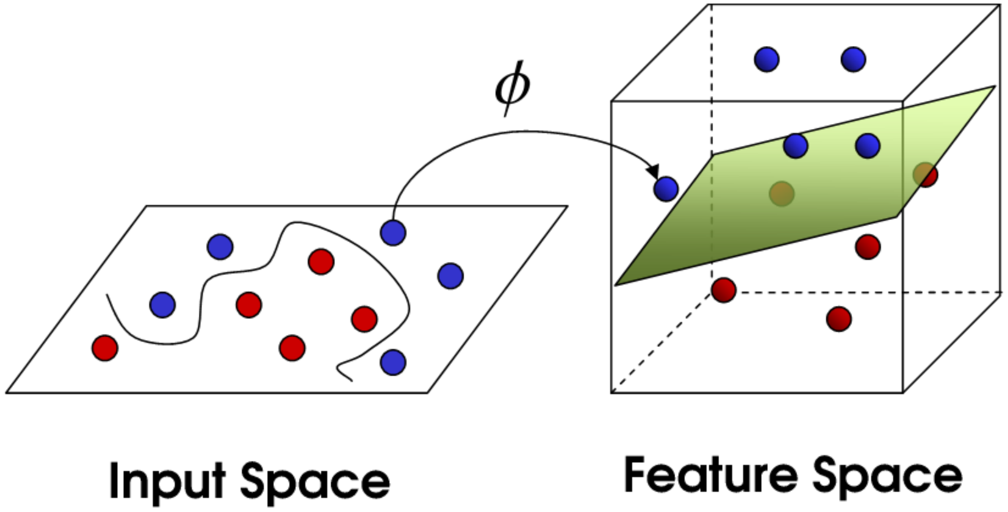
- [Google Colab notebook]()



Kernel PCA

Kernel PCA

- PCA can be extended to find non-linear "directions"
- Can leverage "kernel trick" to perform PCA in kernel space



$$\Sigma_\phi \boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1$$

$$\Sigma_\phi = \frac{1}{n} \sum_{i=1}^{n} \phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)^T$$

Image from: D. Wilimitis, The Kernel Trick in Support Vector Classification.

Kernel PCA

- Principal component direction in feature space is linear combination of transformed points
- Weight vector, **c**, is the eigenvector corresponding to largest eigen value of the kernel matrix
- Weight vector constraint

$$\boldsymbol{u}_1 = \sum_{i=1}^{n} c_i \phi(\boldsymbol{x}_i)$$

$$\boldsymbol{c} = \left(c_1, c_2, \cdots, c_n\right)^T$$

$$\boldsymbol{K}\boldsymbol{c} = n\lambda_1 \boldsymbol{c} = \eta_1 \boldsymbol{c}$$

$$\|\boldsymbol{c}\|^2 = \frac{1}{\eta_1}$$

Kernel PCA Algorithm

KernelPCA $(\boldsymbol{D}, K, \alpha)$**:**

$\boldsymbol{K} = \{K(\boldsymbol{x}_i, \boldsymbol{x}_j)\}_{i,j=1,\dots,n}$ // compute $n \times n$ kernel matrix

$\boldsymbol{K} = (\boldsymbol{I} - \frac{1}{n}\boldsymbol{1}_{n\times n})\boldsymbol{K}(\boldsymbol{I} - \frac{1}{n}\boldsymbol{1}_{n\times n})$ // center the kernel matrix

$(\eta_1, \eta_2, \dots, \eta_d) = \text{eigenvalues}(\boldsymbol{K})$ // compute eigenvalues

$\begin{pmatrix}\boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_n\end{pmatrix} = \text{eigenvectors}(\boldsymbol{K})$ // compute eigenvectors

$\lambda_i = \frac{\eta_i}{n}$ for all $i = 1, \dots, n$ // compute variance for each component

$\boldsymbol{c}_i = \sqrt{\frac{1}{\eta_i}} \cdot \boldsymbol{c}_i$ for all $i = 1, \dots, n$ // ensure that $\boldsymbol{u}_i^T \boldsymbol{u}_i = 1$

$f(r) = \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$, for all $r = 1, 2, \dots, d$ // fraction of total variance

Choose smallest r so that $f(r) \geq \alpha$ // choose dimensionality

$\boldsymbol{C}_r = \begin{pmatrix}\boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_r\end{pmatrix}$ // reduced basis

$\boldsymbol{A} = \{\boldsymbol{a}_i \mid \boldsymbol{a}_i = \boldsymbol{C}_r^T \boldsymbol{K}_i, \text{for } i = 1, \dots, n\}$ // reduced dimensionality data

PCA vs Kernel PCA

TEXAS A&M UNIVERSITY Engineering

- PCA performs linear transformation (centering, rescaling, and rotation)
- Data is already centered and no rescaling (PCA causes rotation)
- Kernel PCA more effective for non-linearly separable data

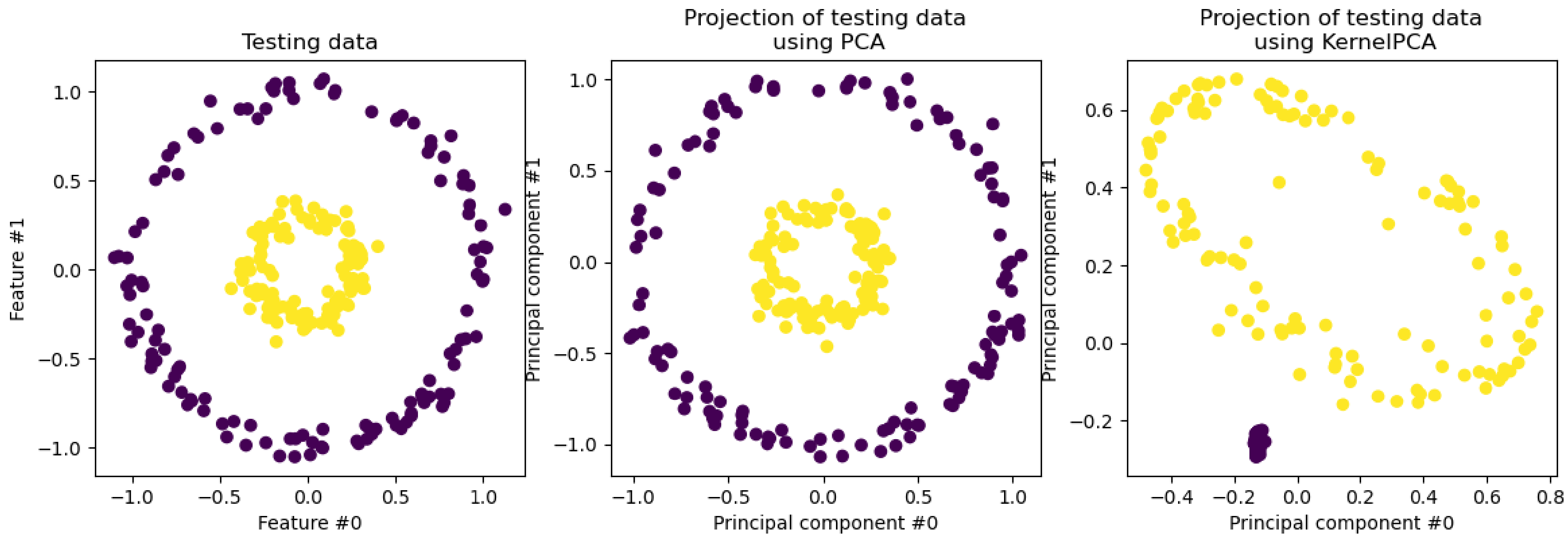


Image from: Sklearn, Kernel PCA.

Singular Value Decomposition

Singular Value Decomposition

- PCA special case of SVD
- Generalizes factorization for any matrix

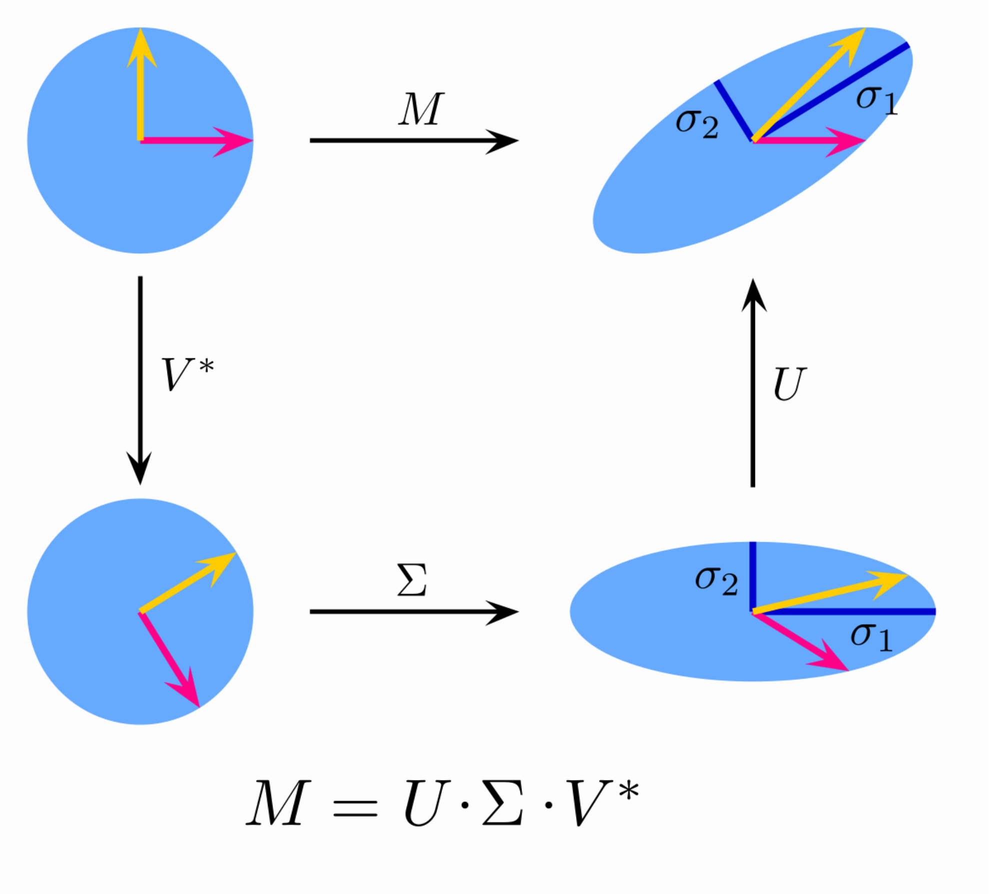
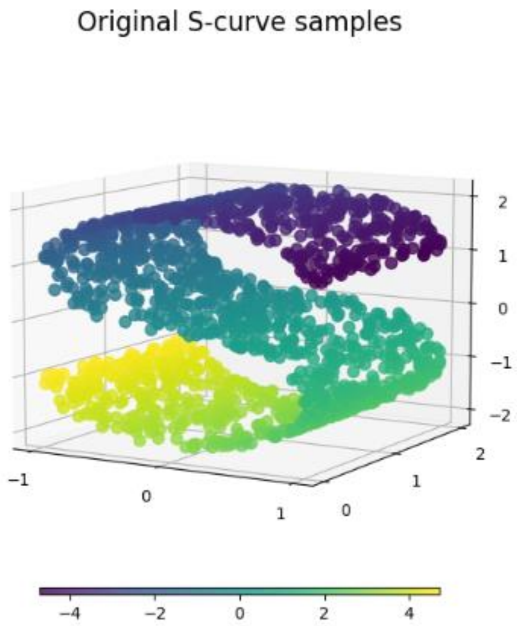


Image from: Wikipedia, Singular Value Decomposition.

Other Dimensionality Reduction Techniques

Sklearn Dimensionality Reduction



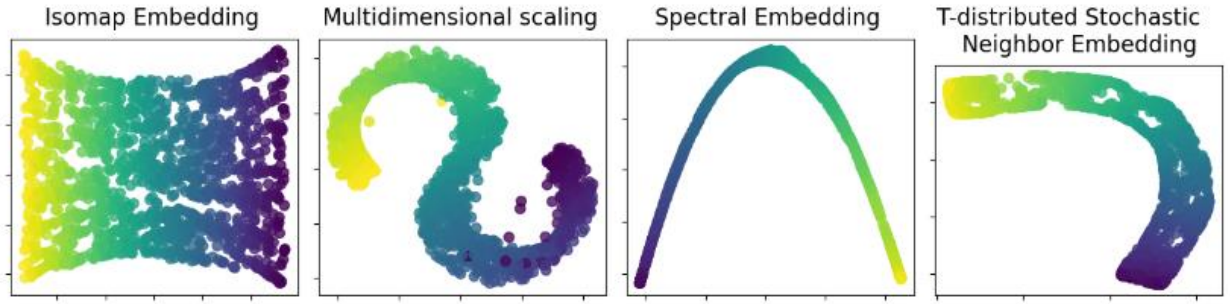


Image from: Sklearn, Unsupervised Dimensionality Reduction.

t-distributed Stochastic Neighbor Embedding

- t-SNE
- Uses joint distribution of higher and lower dimension to model perform dimensionality reduction

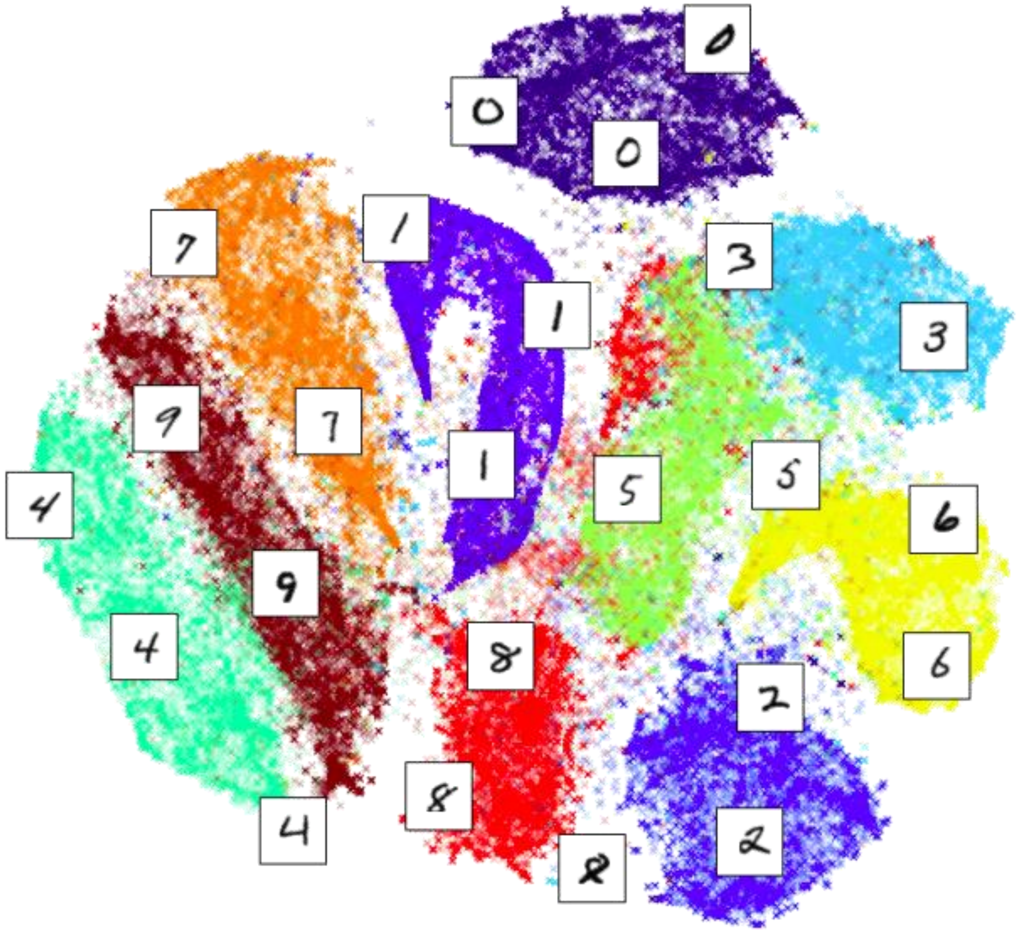
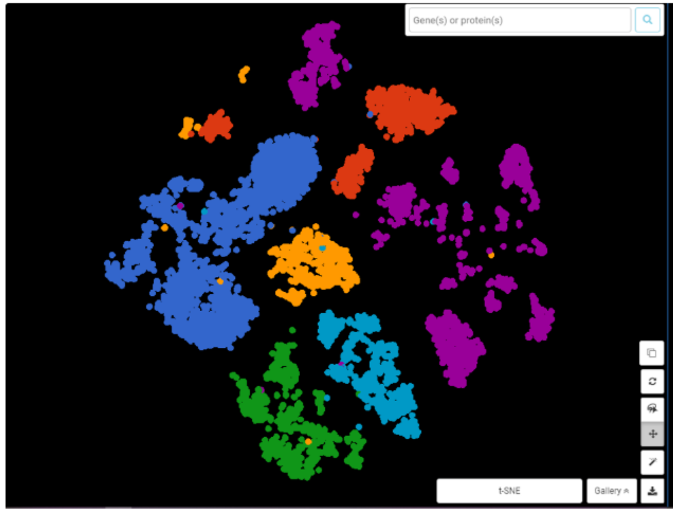


Image from: L. Schoneveld, In Raw Numpy-tSNE.

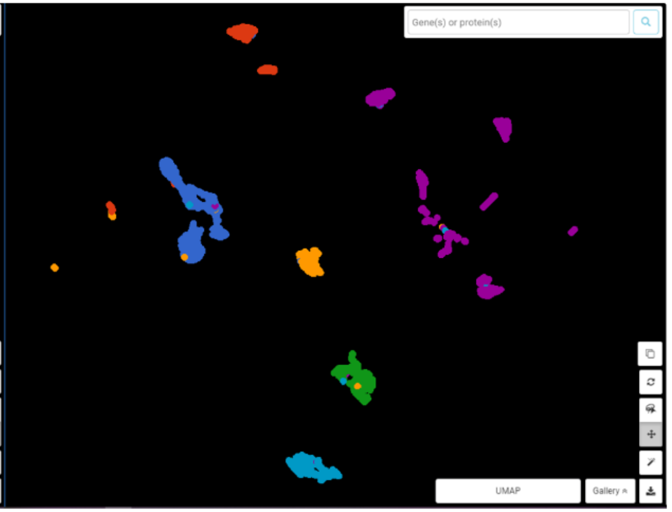
Uniform Manifold Approximation and Projection

TEXAS A&M UNIVERSITY Engineering

- UMAP
- Assumptions
  - Data is uniformly distributed
  - Metric is locally constant
  - Manifold is locally connected



TSNE



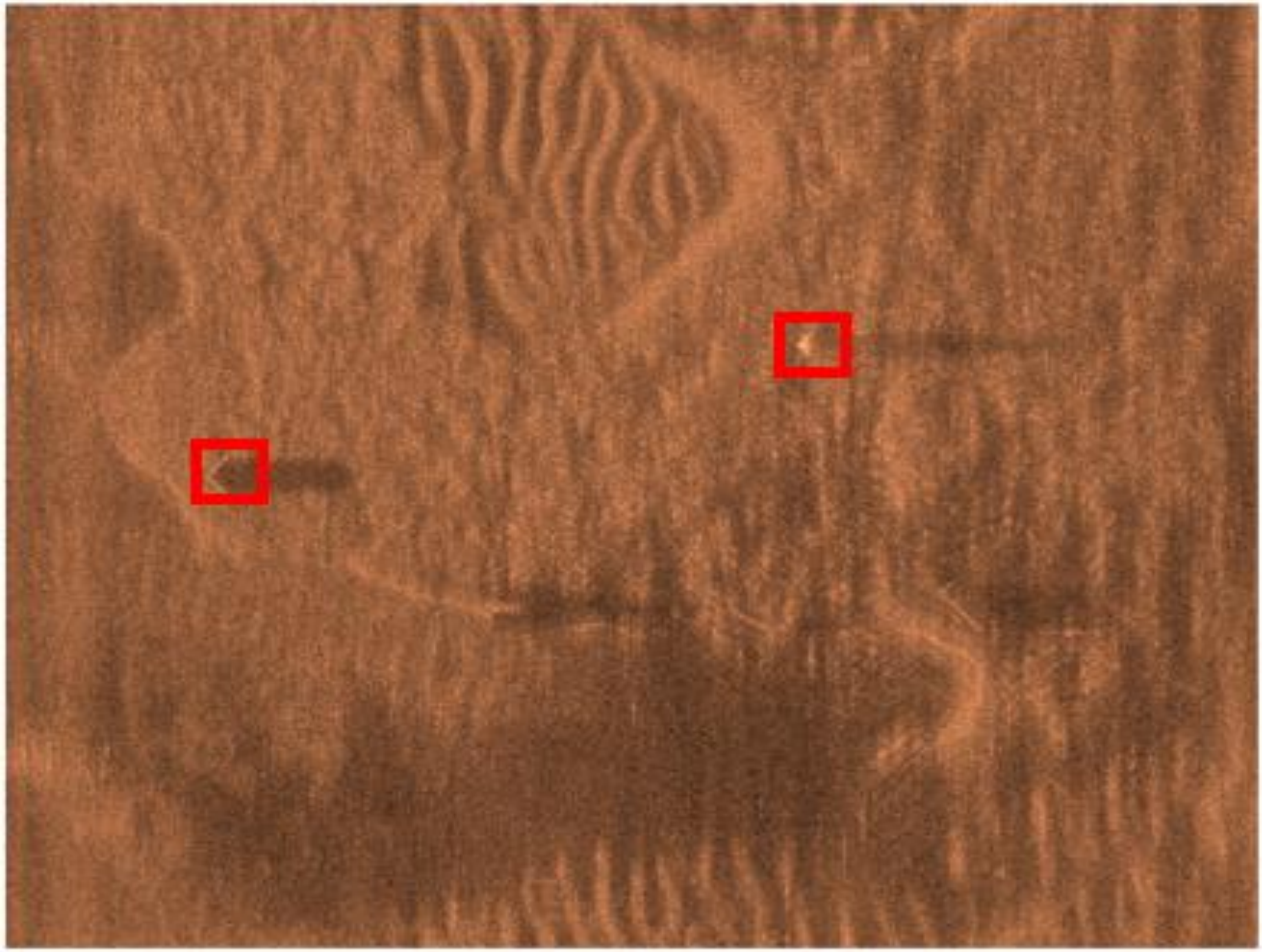
UMAP

Image from: M.H. Tran, Comparing UMAP vs t-SNE in Single-cell RNA-Seq Data Visualization, Simply Explained.

Dimension Reduction Applications

Defense

Automatic Target Recognition



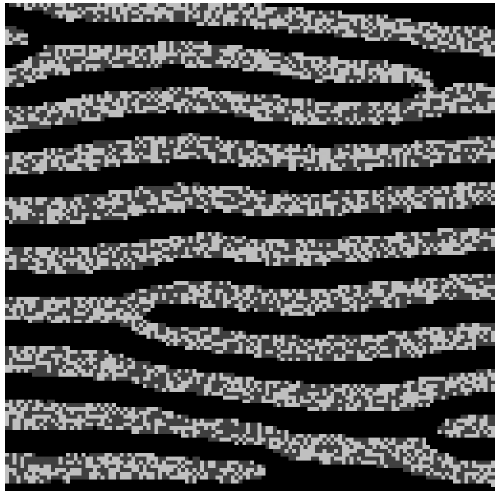
J. **Peeples,** A. Zare, J. Dale, and J. Keller. Histogram Layers for Synthetic Aperture Sonar Imagery, IEEE ICMLA, 2022.

Statistical SAS Images

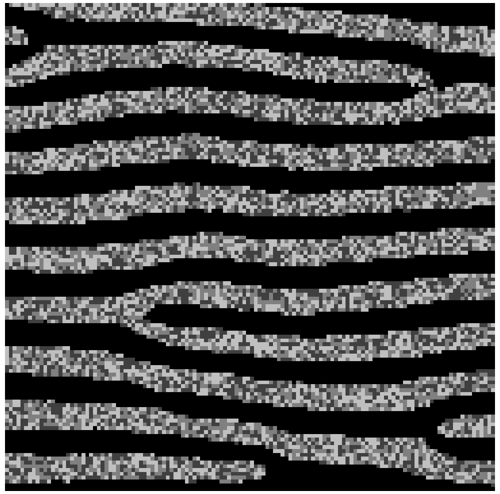
TEXAS A&M UNIVERSITY Engineering

- Created statistical textures using Pseudo Image Synthetic Aperture Sonar (PISAS) dataset
- Two structures: sand ripple and rocky

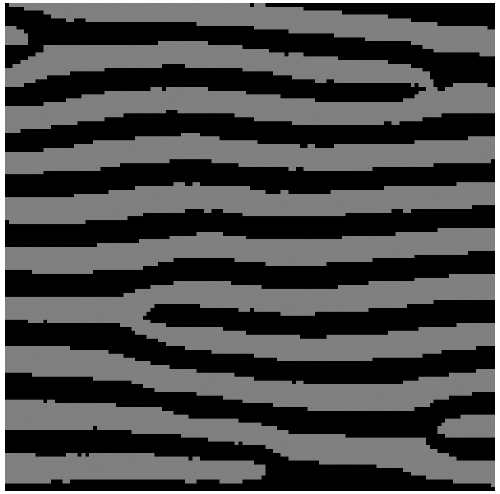
S1: Binomial



S2: Multinomial



S3: Constant

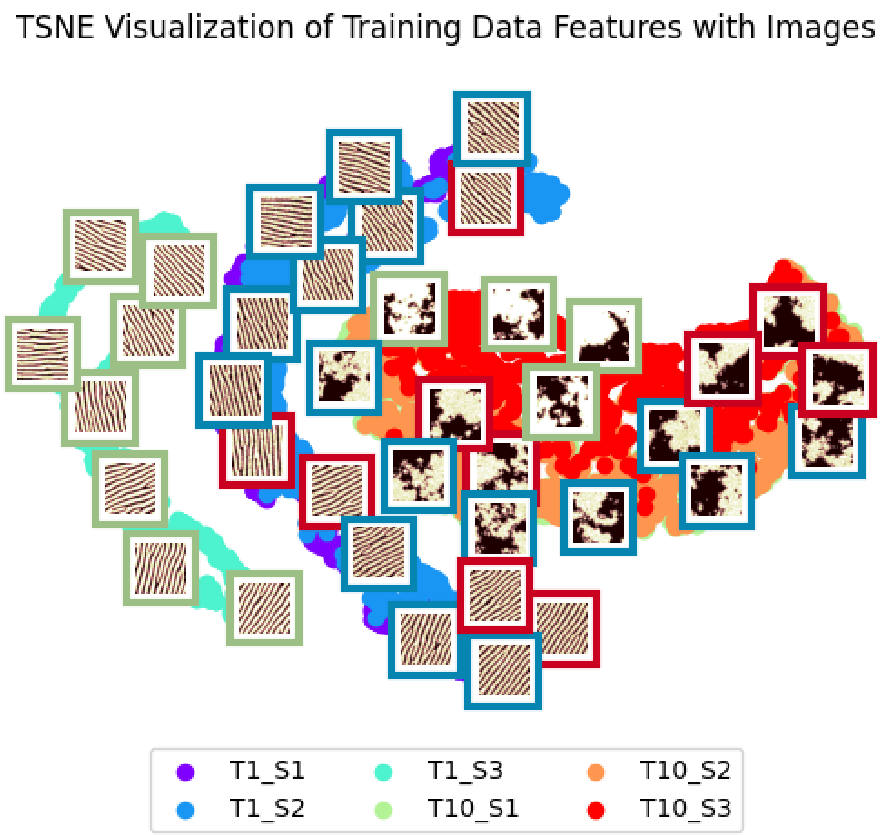


D. Stewart, et. al. Image-to-Height Domain Translation for Synthetic Aperture Sonar, IEEE TGRS, 2023.

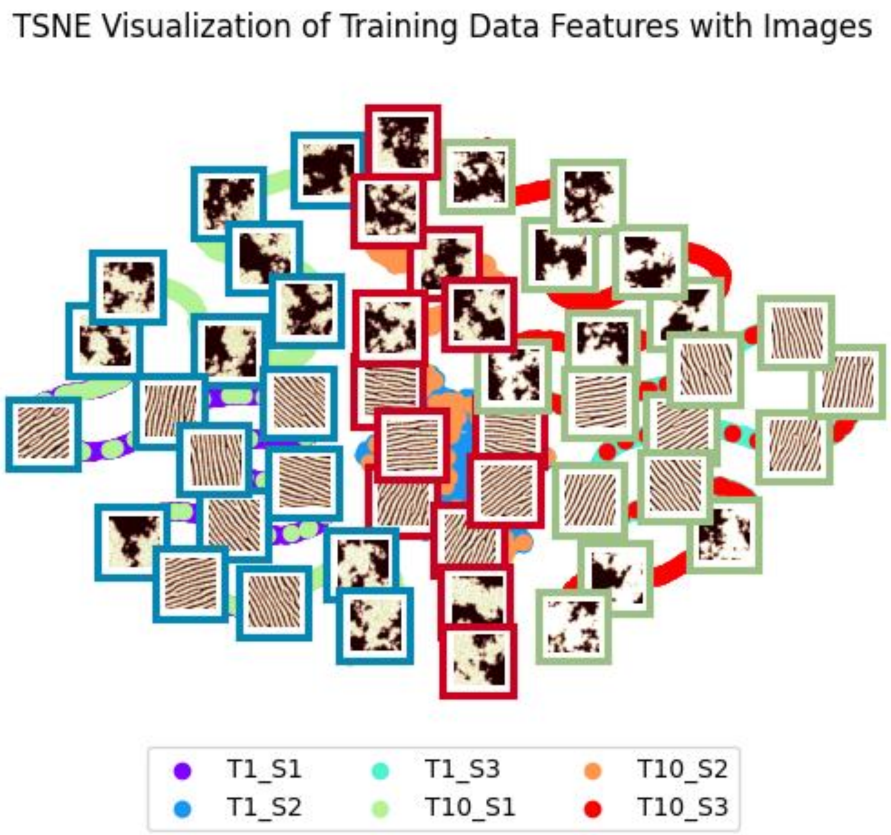
Statistical SAS Images Results

TEXAS A&M UNIVERSITY Engineering

CNN (77.70)

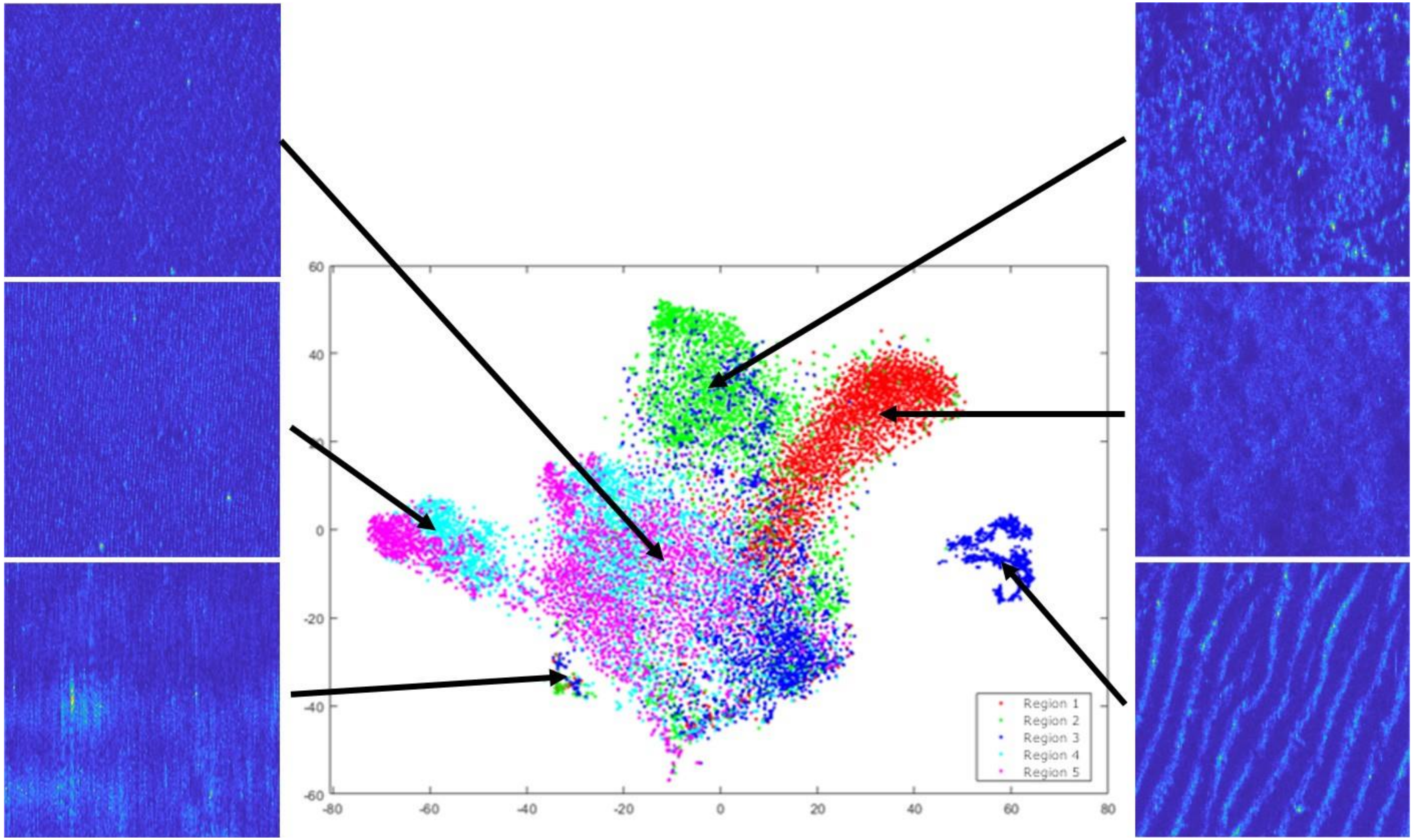


RBF (**82.18**)



J. Peeples, A. Zare, J. Dale, and J. Keller. Histogram Layers for Synthetic Aperture Sonar Imagery, IEEE ICMLA, 2022.

Generalization to SAS Images

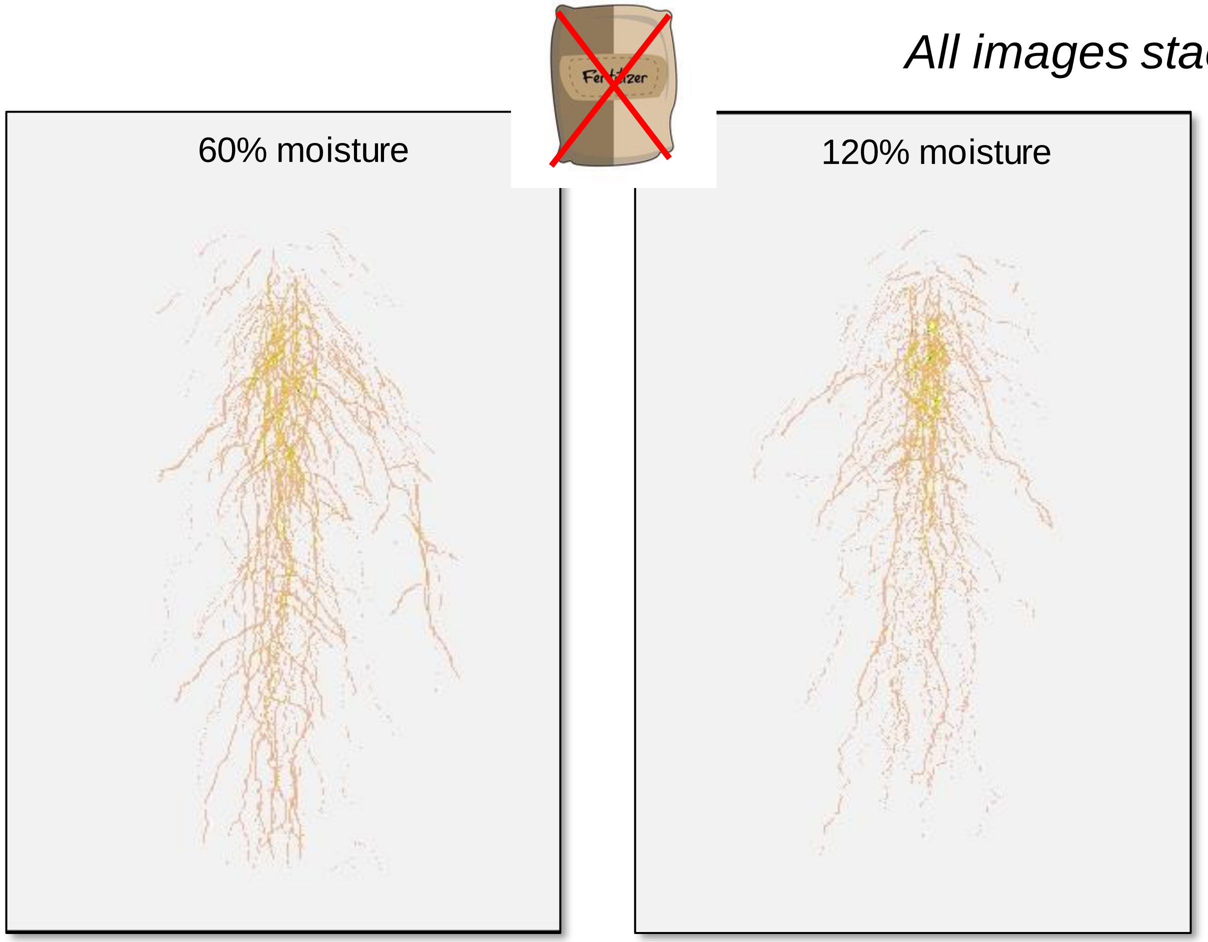


J. Peeples, A. Zare, J. Dale, and J. Keller. Histogram Layers for Synthetic Aperture Sonar Imagery, IEEE ICMLA, 2022.

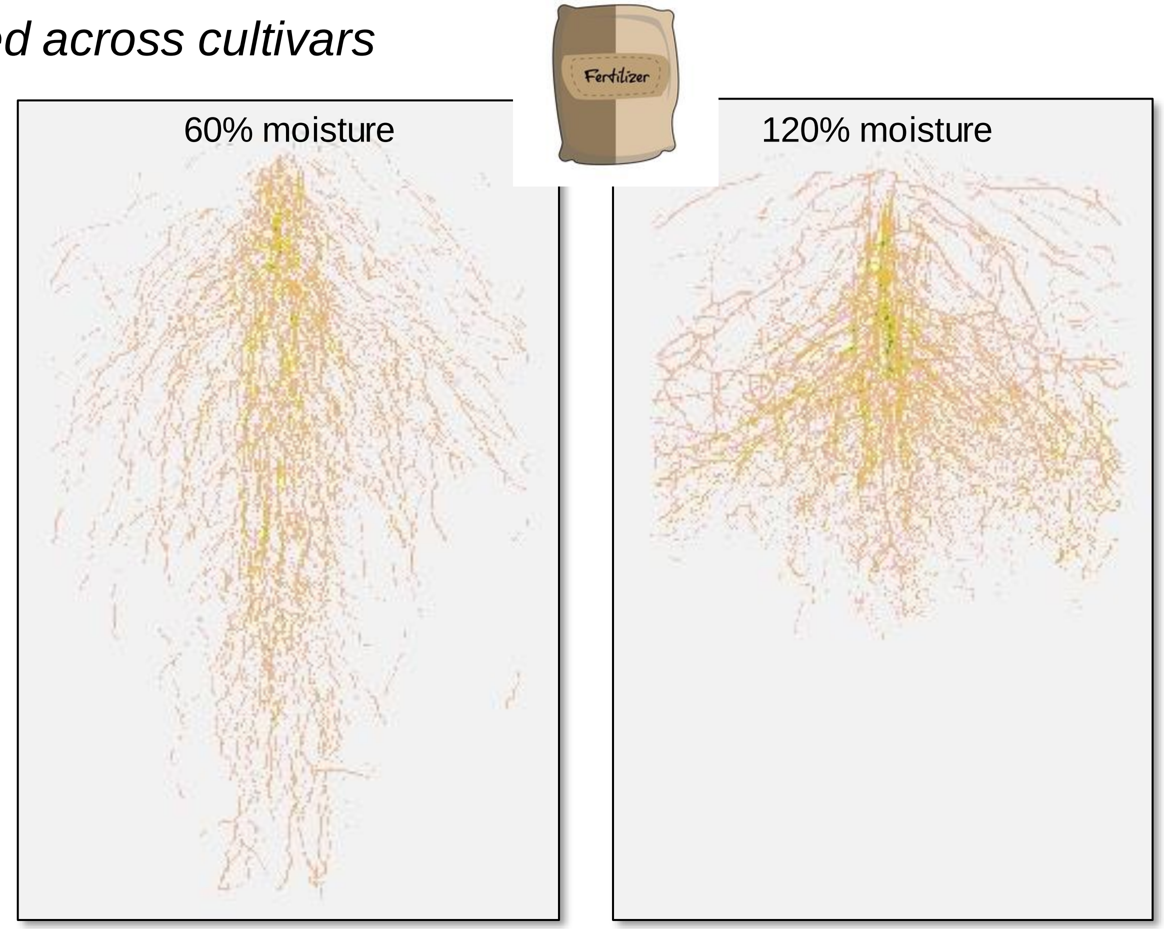
Agriculture

Traditional WinRhizotron Analyses do not capture the whole picture

All images stacked across cultivars



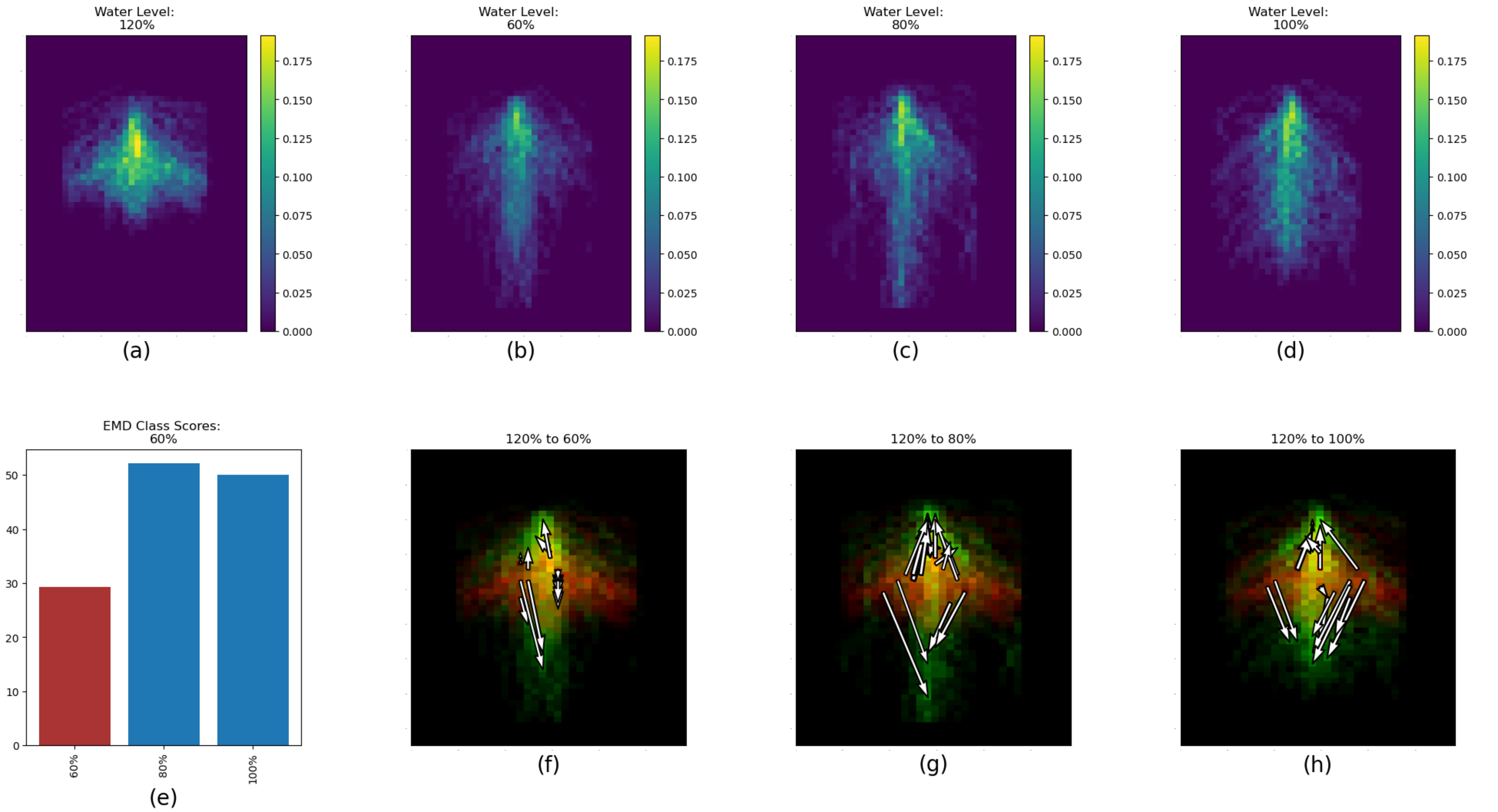
Not enough resources to avoid the flooding stress



Enough resources to produce more lateral roots and avoid the waterlogged soil

J. Peeples, W. Xu, R. Gloaguen, et. al. Spatial and Texture Analysis of Root System Distribution with Earth Mover's Distance (STARSEED), Plant Methods, 2023.

EMD Workflow for Water Level 120% (with fertilizer)

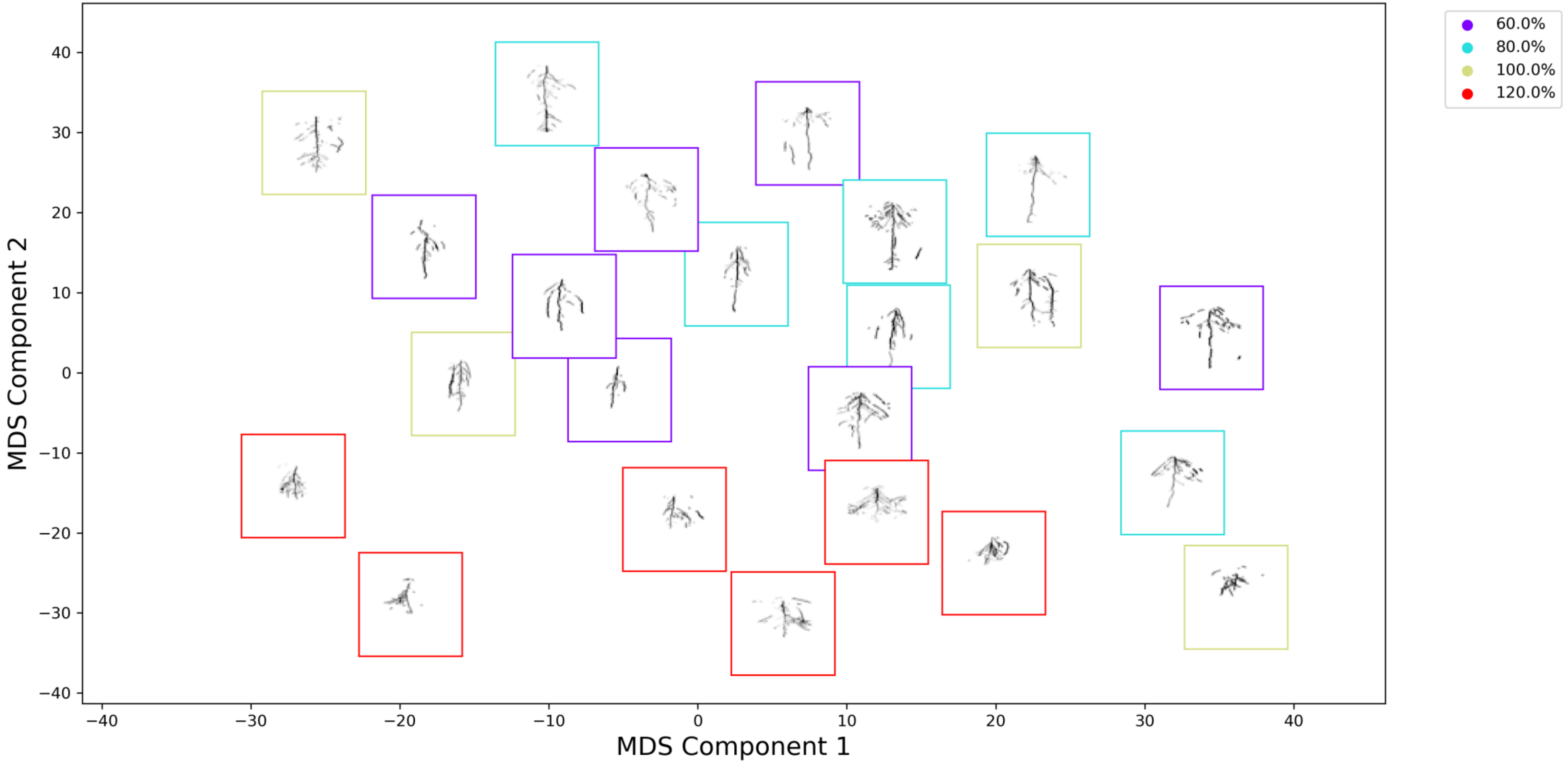


Red channel is current class; green channel is other class; yellow means "unchanged"

J. Peeples, W. Xu, R. Gloaguen, et. al. Spatial and Texture Analysis of Root System Distribution with Earth Mover's Distance (STARSEED), Plant Methods, 2023.

Plant Root Analysis

TEXAS A&M UNIVERSITY Engineering



J. Peeples, W. Xu, R. Gloaguen, et. al. Spatial and Texture Analysis of Root System Distribution with Earth Mover's Distance (STARSEED), Plant Methods, 2023.

Next class

TEXAS A&M UNIVERSITY Engineering

- No class Monday (Labor Day)
- Wednesday (09/04): Frequent Itemset Mining and Association Rules

TEXAS A&M UNIVERSITY Engineering

Thank You! Questions?

Joshua Peeples, Ph.D.

https://www.joshpeeples.com/

jpeeples@tamu.edu

TEXAS A&M UNIVERSITY
Engineering

Supplemental Slides

Dimensionality Reduction Resources

- [Eigenvectors and eigenvalues]
- [PCA in 5 minutes]
- [PCA Step-by-Step]
- [Introduction to KPCA]