Unsupervised Learning

Johanni Brea

Introduction to Machine Learning

Data Generating Processes Revisited

Recap

It is useful to think of our datasets as samples from **data generating processes** for the input X and the conditional output Y|X.

- **MNIST**
  X: people write digits → people take standardized photos thereof.
  Y|X: different people label the same photo X.
- **Weather**
  X: the weather acts on sensors in weather stations.
  Y|X: the weather evolves from X and is measured again 5 hours later.

Using samples from these data generating processes, supervised learning aims at learning something about the conditional processes, i.e how Y depends on X.

Using samples from these data generating processes, **unsupervised learning** aims at learning something about the input generator, i.e how X is generated.

Terminology

- **Supervised Learning**: learn $p(Y|X)$
- **Semi-Supervised Learning**: learn $p(Y|X)$ with typically a small fraction of the data having labels given explicitly by humans and the rest unlabeled, e.g. many images, but only some with labels.
- **Self-Supervised Learning**: learn $p(Y|X)$ where Y is not a label given explicitly by humans (or other supervisors). *Example: auto-regressive models like weather prediction.*
- **Unsupervised Learning**: Do something with X, e.g. learn $p(X)$.
  In unsupervised learning one is often more interested in a hidden representation of the data than in plain fitting of $p(X)$, e.g. if the data seems to be clustered, what is the cluster identity of a given point.
  If X is multidimensional one learns sometimes parts of $p(X)$ in a self-supervised manner, e.g. $p(X) = p(X_1)p(X_2|X_1)$.

Goals of Unsupervised Learning

- **Exploratory Data Analysis**: Is there an informative way to visualize the data? Can we discover subgroups among the variables or among the observations?
- **Data Processing**: Can we separate signal from noise (denoising)? Can we efficiently compress the data?
- **Uncovering Hidden "Causes" of Observations**: Can we uncover hidden structure in the data? Does the data lie on a low-dimensional manifold?
- **Generating Artificial Data**: Can we generate high-quality novel data samples, e.g. images, text or music?

For the assessment of unsupervised learning there are often no clear objective guidelines.

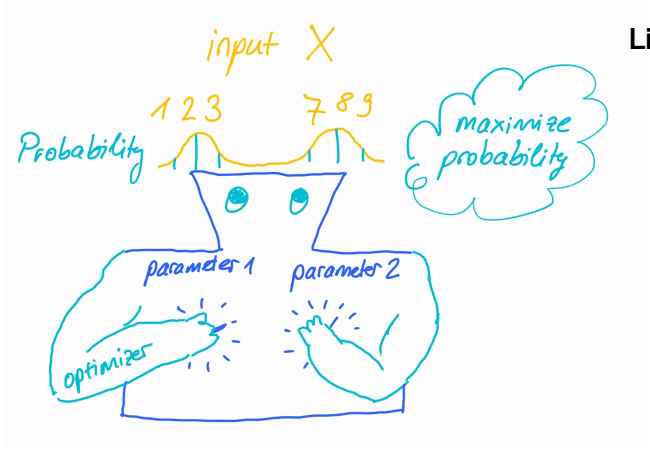
Table of Contents

1. How Does Unsupervised Learning Work?

2. Clustering
   - Gaussian Mixture Models
   - Density-based spatial clustering of applications with noise

3. Dimensionality Reduction
   - Recap PCA
   - Matrix Completion
   - t-SNE
   - UMAP

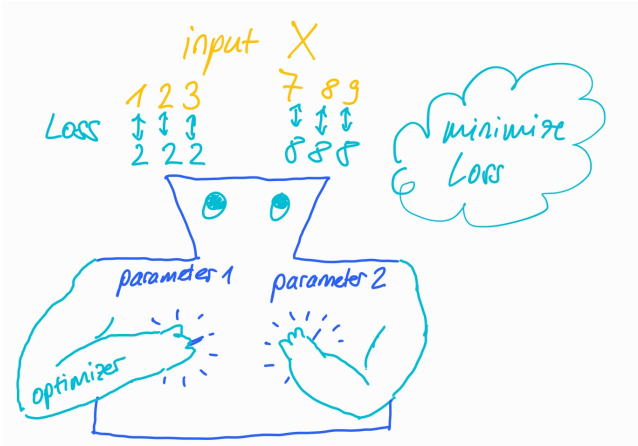
How Does Unsupervised Learning Work?



Likelihood Maximizing Machine

- We specify
  1. the training data
  2. the family of probability distributions (model)
  3. the optimizer
- The machine changes the parameters with the help of the optimizer until the likelihood of the parameters is maximal.

E.g.: Gaussian Mixture Model



Loss Minimizing Machine

- We specify
  1. the training data
  2. the function family (model)
  3. the loss function $L(x)$
  4. the optimizer
- The machine changes the parameters with the help of the optimizer until the loss is minimal.

E.g.: K-Means Clustering
(not further discussed here)

Table of Contents

1. How Does Unsupervised Learning Work?

2. Clustering
   - Gaussian Mixture Models
   - Density-based spatial clustering of applications with noise

3. Dimensionality Reduction
   - Recap PCA
   - Matrix Completion
   - t-SNE
   - UMAP

Gaussian Mixture Models

Assume the data generating process

1. Sample component k with probability ϕ_k ($\sum_{k=1}^{K} \phi_k = 1$)
2. Sample data point $x \in \mathbb{R}^d$ from multivariate normal distribution with mean $\mu_k \in \mathbb{R}^d$ and covariance matrix Σ_k.

$$p(x|\theta) = \sum_{k=1}^{K} \phi_k \mathcal{N}(x; \mu_k, \Sigma_k)$$

parameters $\theta = (\phi_1, \mu_1, \Sigma_1, \ldots, \phi_K, \mu_K, \Sigma_K)$

Given data $(x_1, \ldots, x_n)$, fitting a Gaussian Mixture Model amounts to finding the parameters θ that maximize the log-likelihood

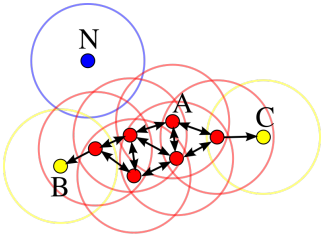
$$\mathcal{L}(\theta) = \sum_{i=1}^{n} \log p(x_i|\theta)$$

Given maximum likelihood parameters $\hat{\theta}$, the posterior probability of data point x_i belonging to component k is proportional to

$$p_{ik} = \hat{\phi}_k \mathcal{N}(x_i; \hat{\mu}_k, \hat{\Sigma}_k)$$

Cluster assignment:
for each i pick k with largest p_{ik}.

Density-based spatial clustering of applications with noise DBSCAN

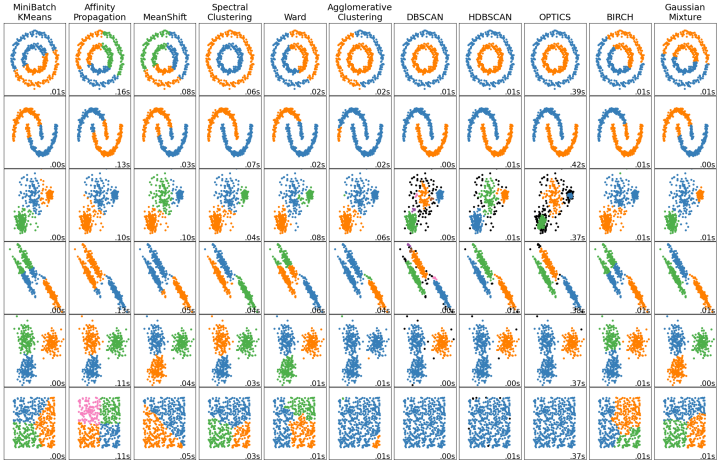


In this diagram, minPts = 4. Point A and the other red points are core points, because the area surrounding these points in an ϵ radius contain at least 4 points (including the point itself). Points B and C are not core points, but are reachable from A (via other core points) and thus belong to the cluster as well. Point N is a noise point that is neither a core point nor directly-reachable.

- minPts and ϵ are hyper-parameters
- No need to predefine the number of clusters
- Robust to noise (outliers)
- Arbitrarily shaped clusters can be found
- HDBSCAN: Newer and hierarchical version of DBSCAN

https://en.wikipedia.org/wiki/DBSCAN

Comparison of Clustering Methods



https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html#sphx-glr-auto-examples-cluster-plot-cluster-comparison-py

Quiz

Correct or wrong?

- Gaussian Mixture Models (GMMs) assume that the data is generated from multivariate Gaussian distributions with different means and identical covariances.
- DBSCAN can find clusters of arbitrary shape and does not require the number of clusters to be specified in advance.
- For a sufficiently large epsilon, DBSCAN merges all data points in one cluster.
- The results of which methods are affected by standardization of the data?
  A) DBSCAN B) GMMs C) K-Means D) Hierarchical Clustering
- Hierarchical clustering always produces the same clusters regardless of the linkage criterion used.
- K-means clustering is guaranteed to find the global optimum clustering solution for any dataset.

Table of Contents

1. How Does Unsupervised Learning Work?

2. Clustering
   - Gaussian Mixture Models
   - Density-based spatial clustering of applications with noise

3. Dimensionality Reduction
   - Recap PCA
   - Matrix Completion
   - t-SNE
   - UMAP

First Principal Component

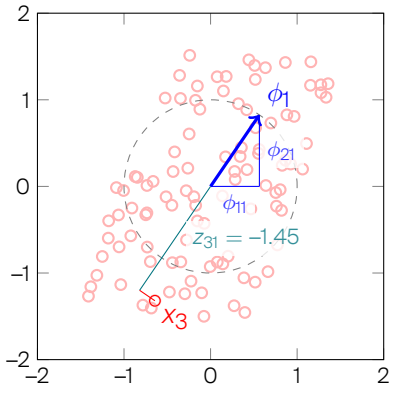
Given x_{ij} for $i = 1, \ldots, n, j = 1, \ldots, p$
column-wise zero mean $\bar{x}_j = \frac{1}{n} \sum_{i=1}^{n} x_{ij} = 0$.

First Principal Component
Find the direction onto which the projection of the data has the highest variance.

(projection) **scores** $z_{i1} = \langle \phi_1, x_i \rangle = \sum_{j=1}^{p} x_{ij} \phi_{j1}$
Find **loadings** $\phi_{11}, \phi_{21}, \ldots, \phi_{p1}$ that

$$\underset{\phi_{11}, \ldots, \phi_{p1}}{\text{maximize}} \frac{1}{n} \sum_{i=1}^{n} z_{i1}^2$$

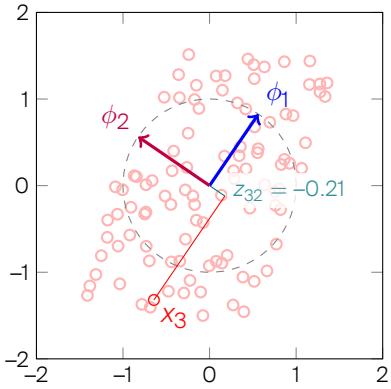
under the constraint $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.



Principal Component Analysis

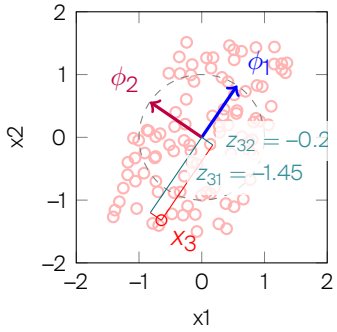
Second Principal Component
Find the direction onto which the projection of the data has the highest variance under the constraint that it is orthogonal to the first PC.

k-th Principal Component Find the direction onto which the projection of the data has the highest variance under the constraint that it is orthogonal to the first $k-1$ PCs.

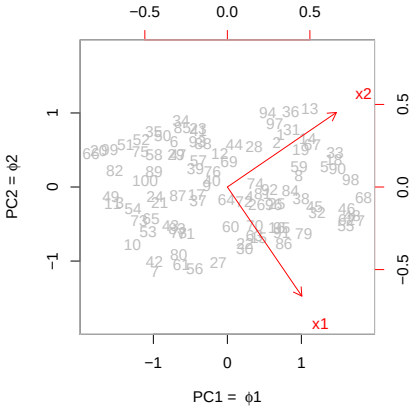


Biplots: Visualizing Scores and Loadings Simultaneously

Plot of raw data



Biplot



scores in gray: read bottom-left coordinates,
loadings in red: read top-right coordinates. (read the coordinates of the text label, e.g. x1, not the arrow tip).

First Principal Component Rewritten in Matrix Notation

$$z_1 = X\phi_1$$

data	X	$n \times p$ matrix	row i contains observation i
loadings	ϕ_1	$p \times 1$ column vector	first PC
scores	z_1	$n \times 1$ column vector	first PC scores of all observation

$$\text{Find } \phi_1 \text{ that } \underset{\|\phi_1\|=1}{\text{maximizes}} \frac{1}{n} z_1^T z_1 = \underset{\|\phi_1\|=1}{\text{maximizes}} \frac{1}{n} \phi_1^T X^T X \phi_1$$

The solution of this optimization problem is the eigenvector of the largest eigenvalue of the covariance matrix $X^T X$.

PCA in Matrix Notation and Relation to SVD

$$Z = X\Phi$$

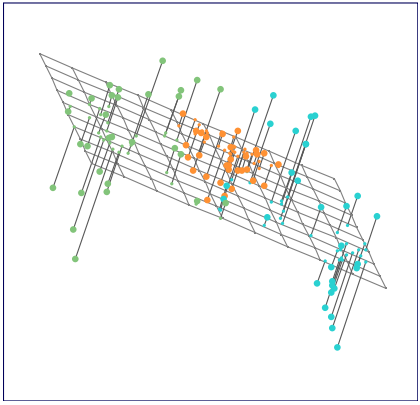
data	X	$n \times p$ matrix	row i contains observation i
loadings	Φ	$p \times p$ matrix	column j contains PC j
scores	Z	$n \times p$ matrix	column j = scores of PC j for all observations

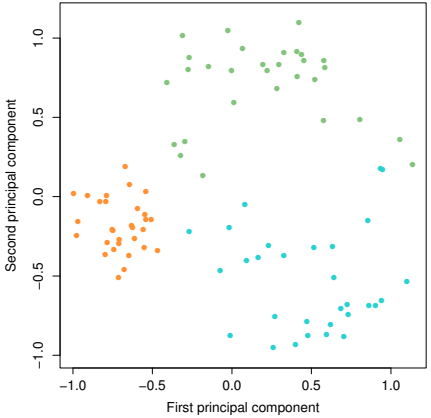
The columns of the loading matrix Φ are eigenvectors of $X^T X$, i.e. $X^T X \phi_i = \lambda_i \phi_i$ or $X^T X \Phi = \Phi \Lambda$ with $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_p)$.

PCA is closely linked to **Singular Value Decomposition** $X = U \Sigma V^T$ where U and V are unitary matrices and Σ is a diagonal matrix. One can show that (see exercises)

$$Z = U\Sigma \quad \text{and} \quad \Phi = V$$

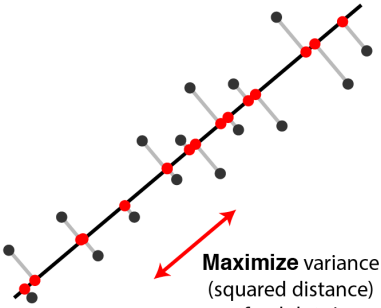
PCA Provides Linear Subspaces Closest to the Data



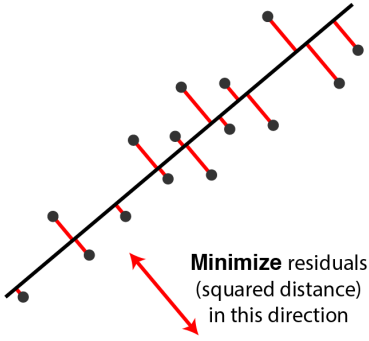


The data varies more within the plane than perpendicular to the plane.

Connection Between the Two Interpretations of PCA



Maximize variance (squared distance) of red dots in this direction



Minimize residuals (squared distance) in this direction

Reinterpretation of Loadings and Scores

Lossless transformation: change of basis

standard basis	$\mathbb{I}$	$p \times p$	columns are standard basis vectors
data	X	$n \times p$	rows are coordinates of points in standard basis
loadings	Φ	$p \times p$	columns are new basis vectors
scores	$Z = X\Phi$	$n \times p$	rows are coordinates of points in new basis
reconstruction	$X = Z\Phi^T$	$n \times p$	rows are coordinates of points in standard basis

Lossy transformation: projection to lower-dimensional space

loadings	Φ_L	$p \times L$	$L = 1, \ldots, p$ first columns of Φ are new basis vectors
scores	$Z_L = X\Phi_L$	$n \times L$	rows are coordinates of points in new basis
reconstruction	$X_L = Z_L\Phi_L^T$	$n \times p$	reconstructed coordinates in standard basis

Φ_L minimizes $\|X - X\Phi_L\Phi_L^T\|_2^2$.

Matrix Completion

	Jerry Maguire	Oceans	Road to Perdition	A Fortunate Man	Catch Me If You Can	Driving Miss Daisy	The Two Popes	The Laundromat	Code 8	The Social Network	⋱
Customer 1	•	•	•	•	4	•	•	•	•	•	⋯
Customer 2	•	•	3	•	•	•	3	•	•	3	⋯
Customer 3	•	2	•	4	•	•	•	•	2	•	⋯
Customer 4	3	•	•	•	•	•	•	•	•	•	⋯
Customer 5	5	1	•	•	4	•	•	•	•	•	⋯
Customer 6	•	•	•	•	•	2	4	•	•	•	⋯
Customer 7	•	•	5	•	•	•	•	3	•	•	⋯
Customer 8	•	•	•	•	•	•	•	•	•	•	⋯
Customer 9	3	•	•	•	5	•	•	1	•	•	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

TABLE 12.2. *Excerpt of the Netflix movie rating data. The movies are rated from 1 (worst) to 5 (best). The symbol • represents a missing value: a movie that was not rated by the corresponding customer.*

- Netflix Prize (competition from 2007 to 2009): recommender system for movies
- Netflix provided a training data set of 100'480'507 ratings that $n = 480'189$ users gave to $p = 17'770$ movies (99% missing values).
- Task: predict missing ratings.
- Can we do better than median imputation?

Dimensionality Reduction

Algorithm 12.1 *Iterative Algorithm for Matrix Completion*

1. Create a complete data matrix $\tilde{\mathbf{X}}$ of dimension $n \times p$ of which the (i, j) element equals

$$\tilde{x}_{ij} = \begin{cases} x_{ij} & \text{if } (i,j) \in \mathcal{O} \\ \bar{x}_j & \text{if } (i,j) \notin \mathcal{O}, \end{cases}$$

where $\bar{x}_j$ is the average of the observed values for the jth variable in the incomplete data matrix $\mathbf{X}$. Here, $\mathcal{O}$ indexes the observations that are observed in $\mathbf{X}$.

2. Repeat steps (a)–(c) until the objective (12.14) fails to decrease:

   (a) Solve

$$\underset{\mathbf{A}\in\mathbb{R}^{n\times M},\mathbf{B}\in\mathbb{R}^{p\times M}}{\text{minimize}} \left\{ \sum_{j=1}^{p}\sum_{i=1}^{n} \left(\tilde{x}_{ij} - \sum_{m=1}^{M} a_{im} b_{jm} \right)^2 \right\} \quad (12.13)$$

   by computing the principal components of $\tilde{\mathbf{X}}$.

   (b) For each element $(i, j) \notin \mathcal{O}$, set $\tilde{x}_{ij} \leftarrow \sum_{m=1}^{M} \hat{a}_{im}\hat{b}_{jm}$.

   (c) Compute the objective

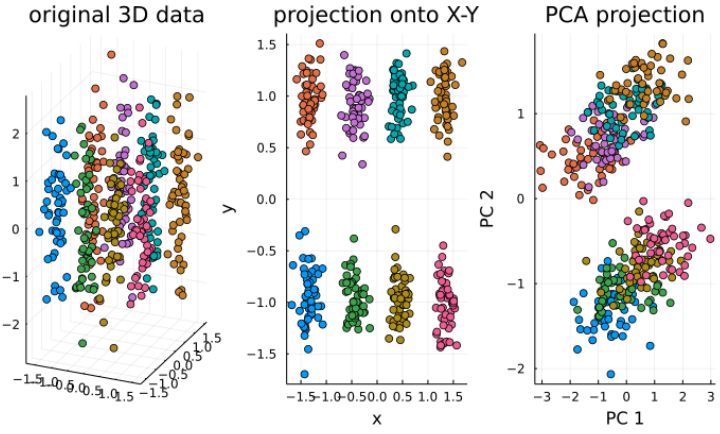
$$\sum_{(i,j)\in\mathcal{O}} \left(x_{ij} - \sum_{m=1}^{M} \hat{a}_{im}\hat{b}_{jm} \right)^2. \quad (12.14)$$

3. Return the estimated missing entries $\tilde{x}_{ij}$, $(i, j) \notin \mathcal{O}$.

- The loss in 12.13 is minimized by $AB^\top = \tilde{X}\Phi_M\Phi_M^\top$
- $AB^\top$ is also called a low-rank (rank M) approximation of the data $\tilde{X}$.
- This is not a silver bullet for dealing with missing data! In a supervised learning: always check your preprocessing, e.g. with cross-validation.

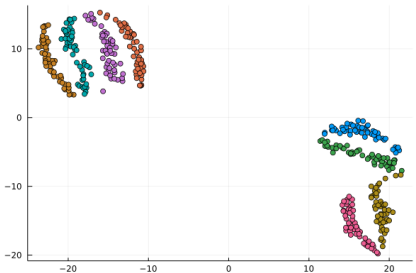
Limitations of PCA

Neighbourhood-relationships can get lost in PCA due to projections.



Alternative 1: t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE)

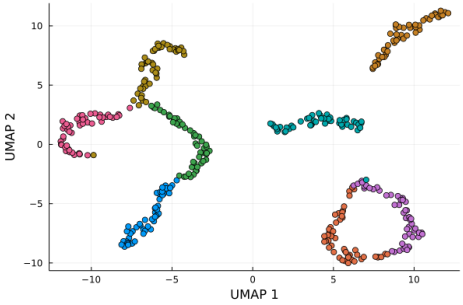


t-SNE tries to preserve local neighbourhood-relationships, but the global structure disappears.

- t-SNE is a gradient descent procedure to find low-dimensional coordinates with the property that the probabilities of being neighbours in the low-dimensional space roughly match the probabilities of being neighbours in the high-dimensional space.
- Hyper-parameters:
  perplexity (smooth measure of num of neighbors)
  number of gradient descent steps.
  dimensionality of lower-dimensional space
- `https://en.wikipedia.org/wiki/T-distributed_stochastic_neighbor_embedding`
  `https://lvdmaaten.github.io/tsne`
  `https://distill.pub/2016/misread-tsne`

Alternative 2: UMAP

Uniform Manifold Approximation and Projection (UMAP)



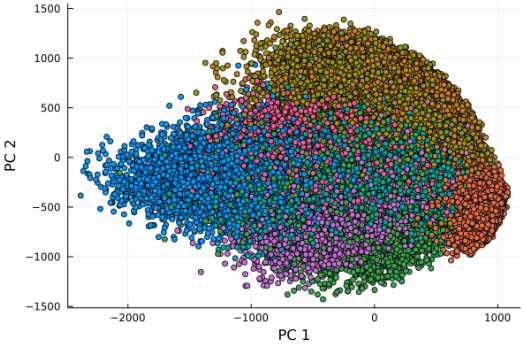
UMAP also tries to preserve local neighbourhood-relationships, but the global structure disappears.

- It is usually faster than t-SNE, and apparently preferable for embeddings in higher dimensions than 2-3.
- Hyper-parameters:
  number of neighbors,
  minimal distance,
  dimensionality of lower-dimensional space,
  metric
- `https://github.com/lmcinnes/umap`
  `https://umap-learn.readthedocs.io/en/latest/parameters.html`

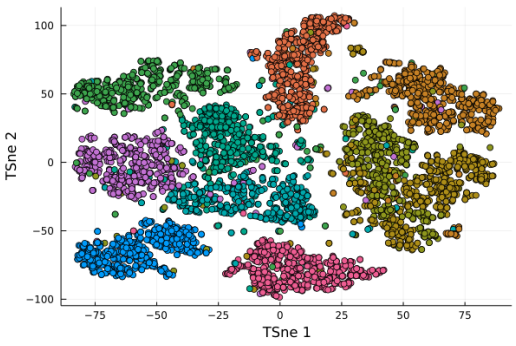
Limitations of PCA

Also in real data there may be clusters that cannot be seen in the first few PCs.

PCA on MNIST images

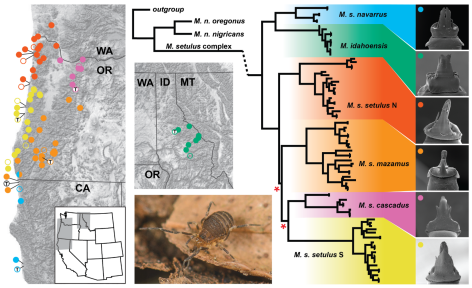


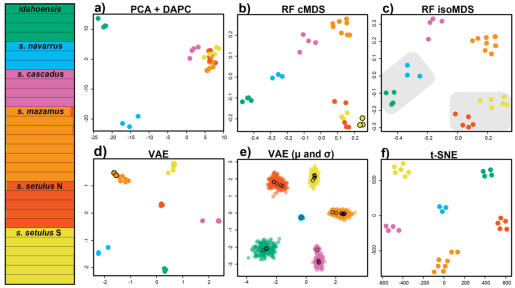
t-SNE on MNIST images



Species Delimitation with Unsupervised Machine Learning

Species Delimitation based on SNP data, dimensionality reduction and clustering:
if different methods find the same clusters, we can be somewhat confident
that the individual clusters are meaningful.





A demonstration of unsupervised machine learning in species delimitation, Derkarabetian et al. 2019

Suggested Reading

- 12 Unsupervised Learning

Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani