Mini Course 3:
Machine Learning II

Clustering, Dimensionality Reduction, and Model Evaluation

Today we'll go over

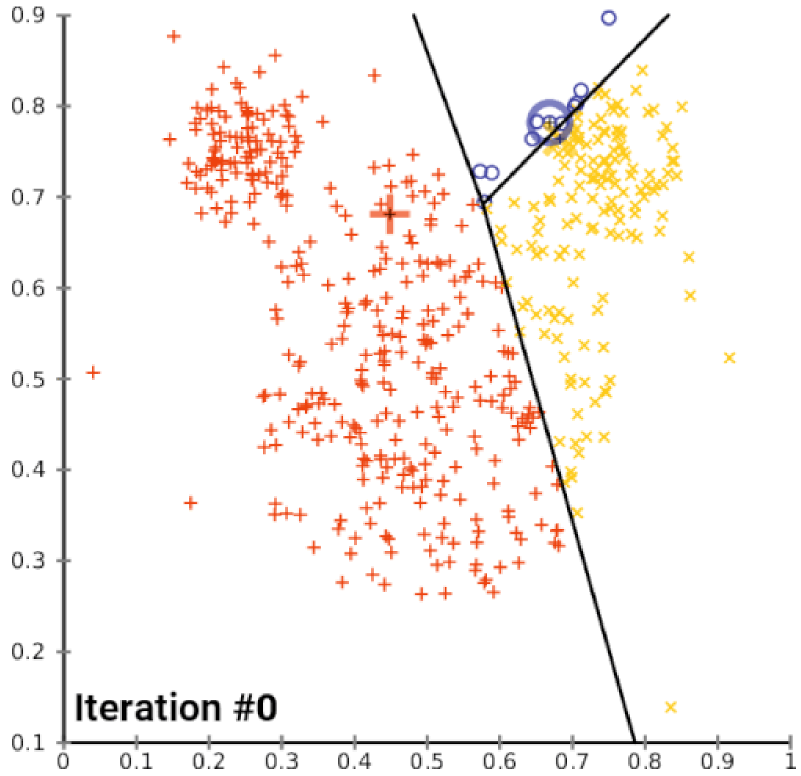
- K-Means Clustering
- Principal Component Analysis
- Hypothesis Testing with ML Models

Unsupervised Machine Learning

- Last week, we learned that supervised machine learning predicts **labels** using **features**
- **Unsupervised learning** learns the *underlying distribution* of features
  - Example: In the automobile data, classes of cars self-organize depending on their features
- A common form of unsupervised learning is **clustering**, which seeks to group similar points based on some criteria

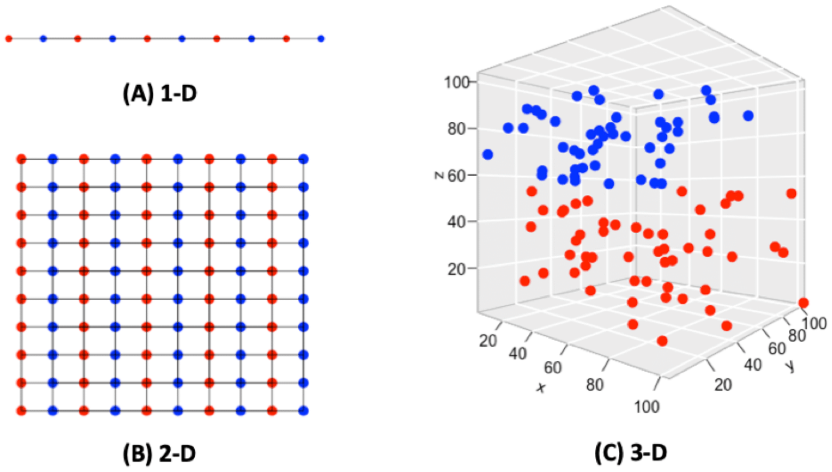
K-Means Clustering

- <u>Goal</u>: Partition dataset into k clusters by finding the closest **centroid** (average of all points in cluster) for each point
  - We decide parameter *k* by some motivated estimate, prior knowledge, or search
  - K-Means assigns points to clusters based on distance to centroids, then recalculates centroids based on new clusters
  - This process repeats until convergence (no change in centroids)
- Notice, no labels are needed to learn to classify the data!



Dimensionality Reduction

- Most data in machine learning research is **high-dimensional** (there are many features in the dataset)
- Sometimes high dimensionality is not ideal
  - Difficult to interpret (can't visualize data to build intuitions)
  - Harder to compute (more data points)
  - The curse of dimensionality – more dimensions means greater distance between data points
- This can strongly influence outcomes of distance-based clustering techniques like K-means
- In this case, we might want to employ **dimensionality reduction**



The Curse of Dimensionality

https://pianalytix.com/k-nearest-neighbour/

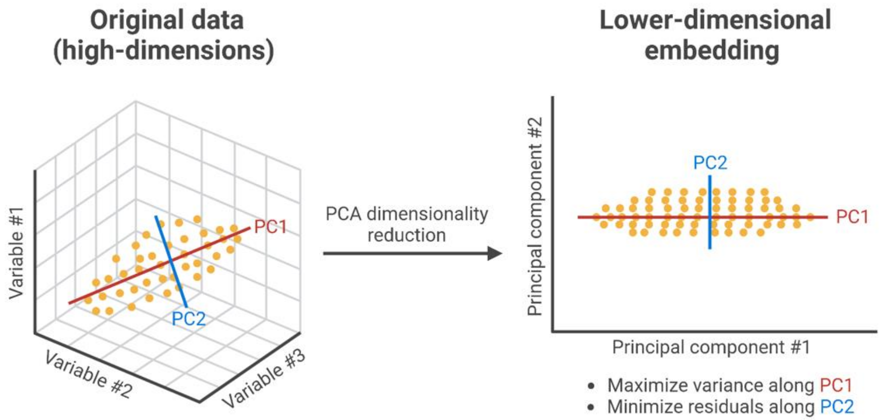
	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	...	engine-size	fuel-system	bore	stroke	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg	price
0	3	NaN	alfa-romero	gas	std	2.0	convertible	rwd	front	88.6	...	130	mpfi	3.47	2.68	9.0	111.0	5000.0	21	27	13495.0
1	3	NaN	alfa-romero	gas	std	2.0	convertible	rwd	front	88.6	...	130	mpfi	3.47	2.68	9.0	111.0	5000.0	21	27	16500.0
2	1	NaN	alfa-romero	gas	std	2.0	hatchback	rwd	front	94.5	...	152	mpfi	2.68	3.47	9.0	154.0	5000.0	19	26	16500.0
3	2	164.0	audi	gas	std	4.0	sedan	fwd	front	99.8	...	109	mpfi	3.19	3.40	10.0	102.0	5500.0	24	30	13950.0
4	2	164.0	audi	gas	std	4.0	sedan	4wd	front	99.4	...	136	mpfi	3.19	3.40	8.0	115.0	5500.0	18	22	17450.0

5 rows × 26 columns

Principal Component Analysis (PCA)

- PCA is a common dimensionality reduction technique that uses <u>matrix factorization</u> to represent the data with fewer features
  - Break dataset into two chunks: components (T), and a weight matrix (W)
  - PCA reduces a dataset of D-dimensions (features) x N-samples to C-dimensions (components) x N-samples

$$\underbrace{T}_{\text{components}} = \underbrace{X}_{\text{original data}} * \underbrace{W}_{\text{weights}}$$

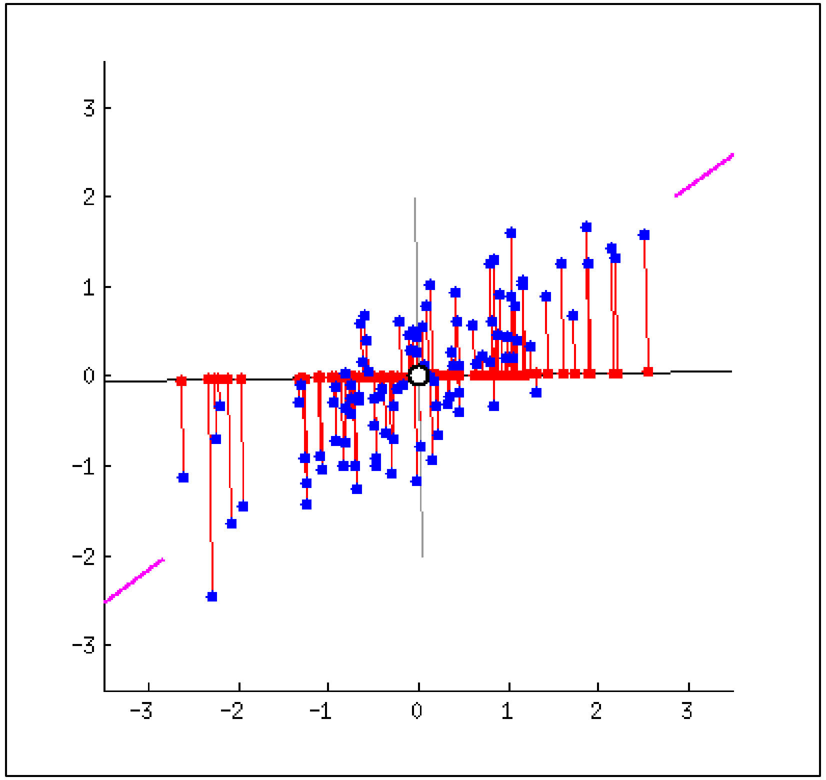


https://www.biorender.com/template/principal-component-analysis-pca-transformation

Principal Component Analysis (PCA) $T = X * W$

- PCA is a common dimensionality reduction technique that uses <u>matrix factorization</u> to represent the data with fewer features
    - Break dataset into two chunks: components (T), and a weight matrix (W)
    - PCA reduces a dataset of D-dimensions (features) x N-samples to C-dimensions (components) x N-samples
    - Goal: Learn W such that variance is maximized, while error is minimized
        - This means that we retain important statistical information about the original data
        - Dimensionality C chosen by parameter search
- Let's take a look at an example

PCA algorithm finds matrix W that:
(1) minimizes distance from blue dots to line
(2) maximizes distance between red dots



https://builtin.com/data-science/step-step-explanation-principal-component-analysis

PCA Example: Sample Dataset

Name	Height (cm)	Weight (kg)	Employed (0=No, 1=Yes)	Birth Year	Death Year
Roald Amundsen	177.42	70.44	0	1872	1928
Harry Mulisch	165.28	98.21	1	1927	2010
Konstantin Melnikov	168.90	76.38	1	1890	1974

Num. Features = 5

Each sample (row) has five elements

High(er) dimensional data

Difficult to visualize full dataset

PCA Example: Reduced Dataset

Name	PC1	PC2
Roald Amundsen	0.021	3.22
Harry Mulisch	0.334	43.09
Konstantin Melnikov	1.22	12.29

Num. Components = 2

Each sample (row) has two principal components

The principal components retain information about the original dataset but in lower dimensionality

We can also plot the whole dataset now that it’s 2D!

A typical pipeline (we’ll try this today!)

- Dimensionality reduction of high-dimensional data
- Clustering on that data
- Visualization in 2D/3D

Hypothesis Testing with ML Models

- Sure, we can train and test a machine learning model, but what are we comparing our results to?
  - What does it mean for a model to have “good” accuracy?
  - In hypothesis testing, it is important to establish a falsifiable statement
    - i.e., reader can infer what you expect to happen, and what would happen if it isn’t true (null hypothesis)
- Hypotheses can be tested by comparing model evaluation to:
  - Statistical baseline
    - Typically, this is chance (random guessing)
    - Could also be something like N standard deviations above chance
  - Other models
    - “Null” model (e.g., a model trained on a shuffled version of an ordered dataset)
    - Models trained on the same task (often common **benchmarks** for specific tasks)
  - Human participants/experts

A Game of Tradeoffs

- The goal of machine learning is to learn the underlying trends in a set of data
  - Often this is to <u>make new predictions from unseen data</u>
- It is important to note that machine learning is partially a game of **tradeoffs**
  - No model is perfect, all have pros and cons!
  - No evaluation method is perfect either!

Jupyter Notebook Time!

- First time?
  - Go to: https://github.com/orbitalhybridization/STARS_ML_MiniCourses
  - Copy the git clone link and run `git clone [url]` in your directory of choice
  - Follow the instructions in mc3/mc3.ipynb
- Returning?
  - Navigate to your directory in VSCode or Terminal
  - Run `git pull` to update your local repository
  - Follow the instructions in mc3/mc3.ipynb

Next week: Reconstructing Visual Percepts from EEG (Simon Fei)

- Simon Fei will discuss his work on using <u>neural networks to reconstruct visual percepts from brain activity</u>
- Regression, dimensionality reduction, clustering – all of these are used in this project!