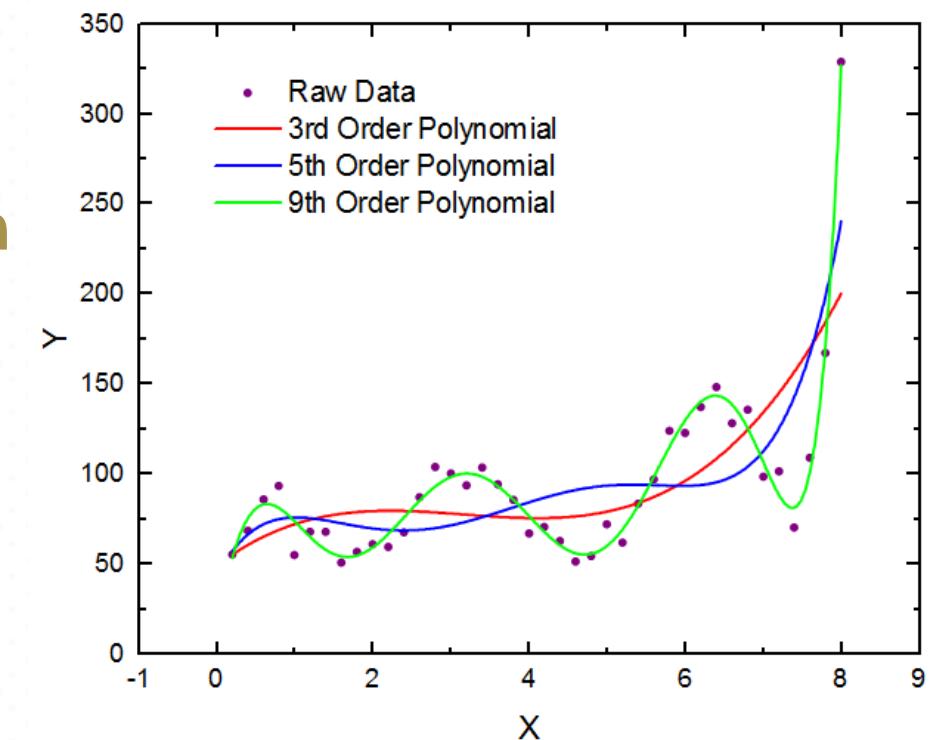


ECE 4803/8803: Fundamentals of Machine Learning (FunML)

Spring 2024

Lecture 10 Part 1: Regression Model Validation



Overview

In This Lecture..

High-degree Polynomial Regression

Training by Gradient Descent

Regularized Regression Models

Performance Measures

Model Validation

- Test/Train/Validation Split
- Learning Curves
- Cross-Validation
- The Bias-Variance Tradeoff

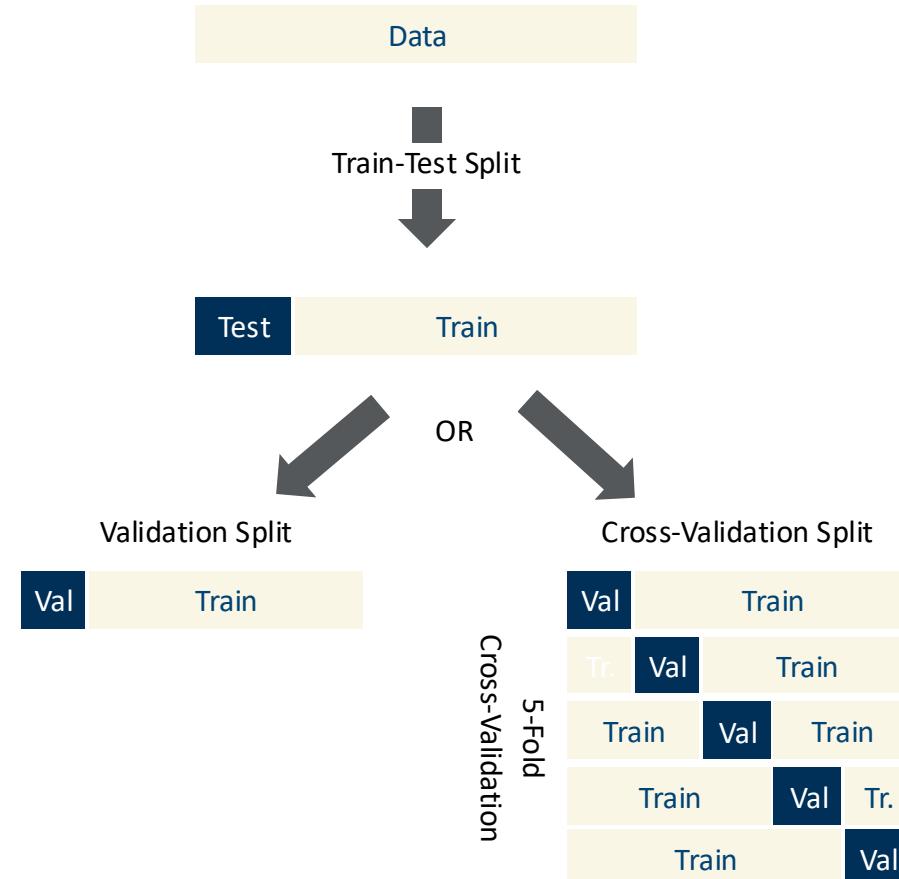
Model Validation

- Training/Validation/Test Split
 - Testing subset kept away and never used during model training
 - Training subset used for training and typically split further into Training/Validation
 - Validation subset for model validation
- Learning Curves
 - Show how *training* and *validation* scores compare as a function of increasing training set size
 - Gives insight on a model's generalization ability
- Cross-Validation
 - Used for assessing how the training results of a model will generalize to unseen data

Model Validation

Train/Test/Validation

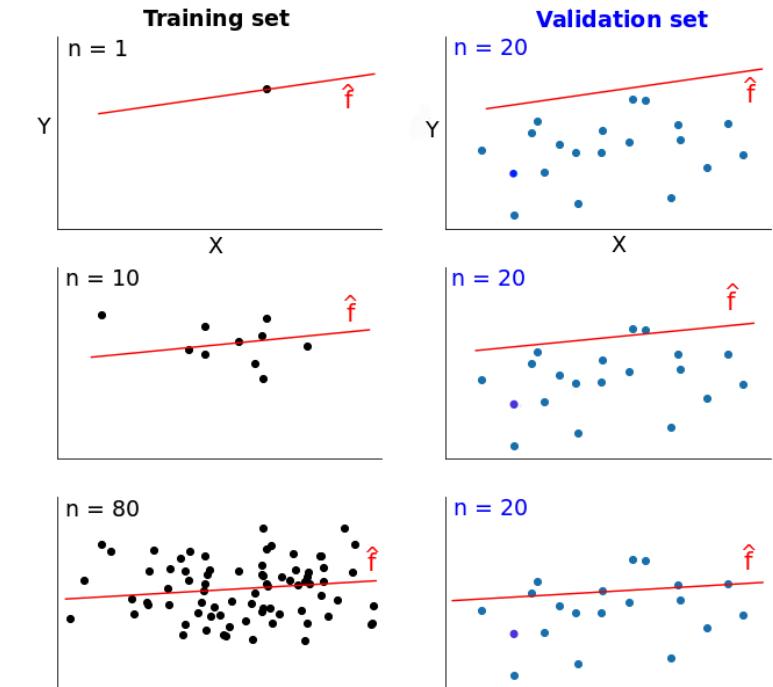
- **Test Dataset**
 - Only used once a model is completely trained (using the train and validation sets)
 - Generally used to evaluate competing models
 - Contains data that spans the various classes in the real world.
- **Training Dataset**
 - The actual dataset used to train the model.
 - The model sees and learns from this data
- **Validation Dataset**
 - Used to evaluate a given model and fine-tune its *hyperparameters*
 - The model sees this data during training but never learns directly from it
 - Typically implemented as cross-validation
- **Split ratio**
 - Typically: %20-30% test, and the remaining training/validation



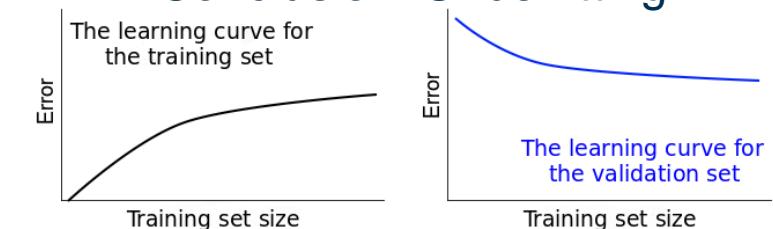
Model Validation

Learning Curves

- Training error starts very low when training set is small, and increases as more training data added
- Validation error starts high and decreases as more training data added.
- The general procedure for generating learning curves is as follows (assuming a dataset of size $n = 100$ samples):
 1. Set aside validation set (e.g., $v = 20$ samples)
 2. For $k = 1$ to $n - v$
 1. Take the first k samples as one training dataset
 2. Fit the model on the training set and evaluate it on the validation set
 3. Retain the training score and the evaluation score and discard the model
 3. Plot the training and evaluation scores recorded in the iterations above against training set sizes



Conclusion: Underfitting

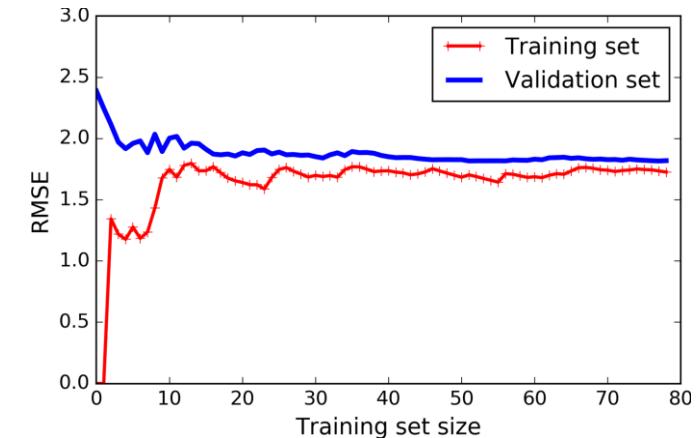


Model Validation

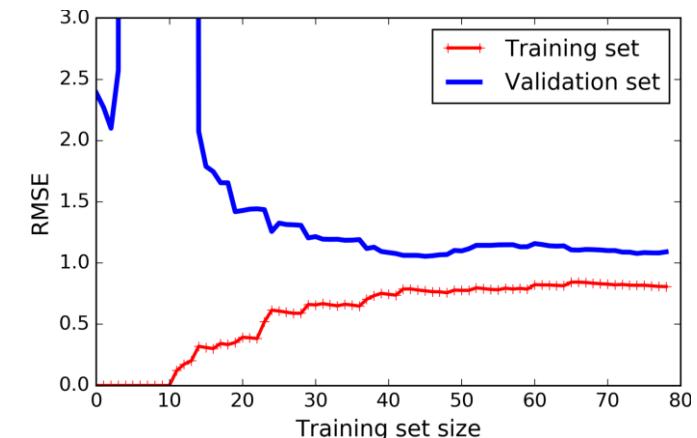
Learning Curves

- Examples of two models trained on the same data.
- Evaluation metric: RMSE
- Linear regression model (top)
 - High error rates
 - Quickly reaches a plateau both in training and validation (underfitting)
- Polynomial regression model
 - Has lower error than linear model,
 - Wider gap between the curves (overfitting)
- How to combat underfitting? Increase complexity of model
- How to combat overfitting? More training data

Linear Regression model



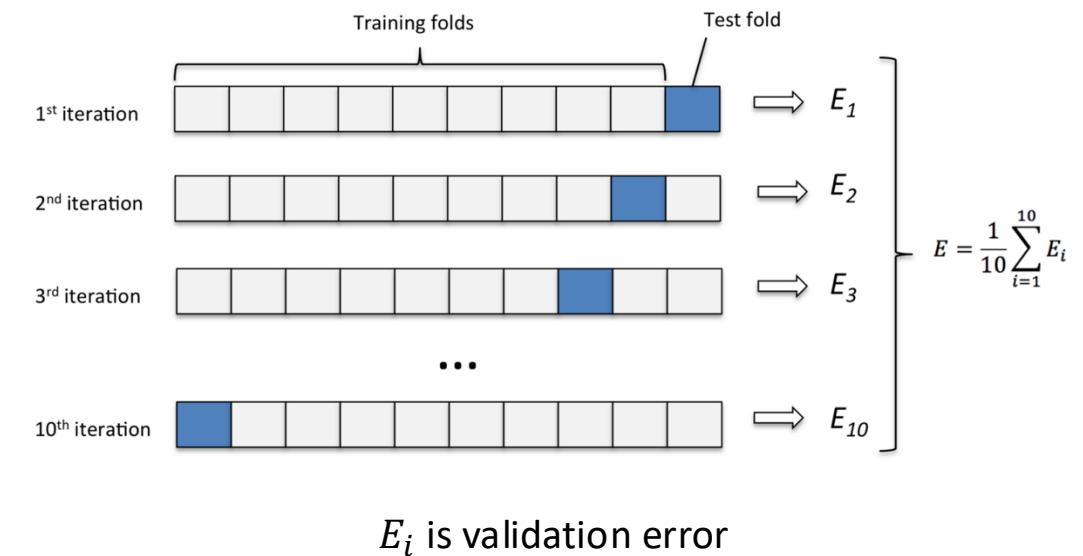
a 10th-degree Polynomial model



Model Validation

Cross Validation

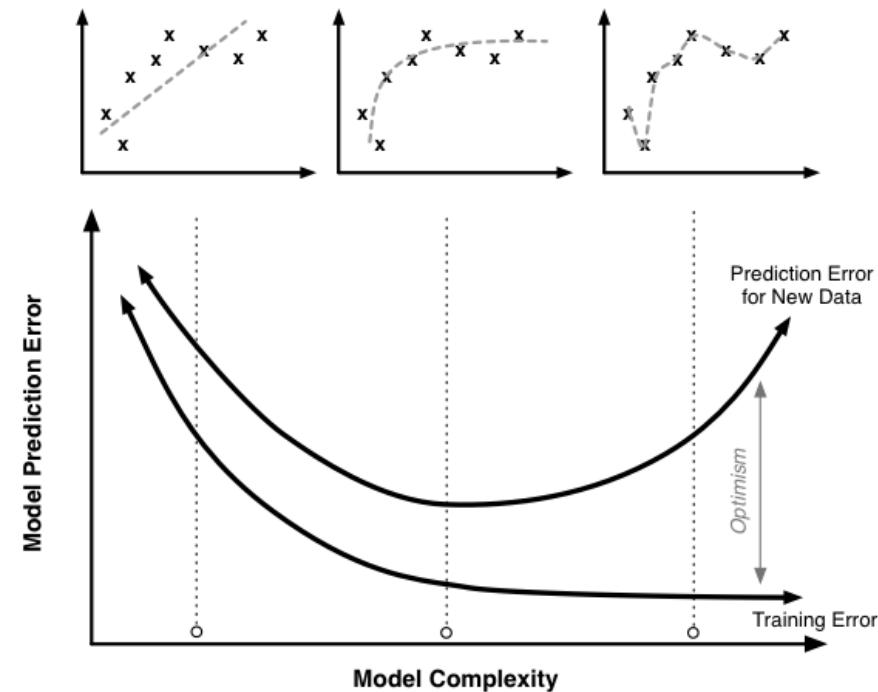
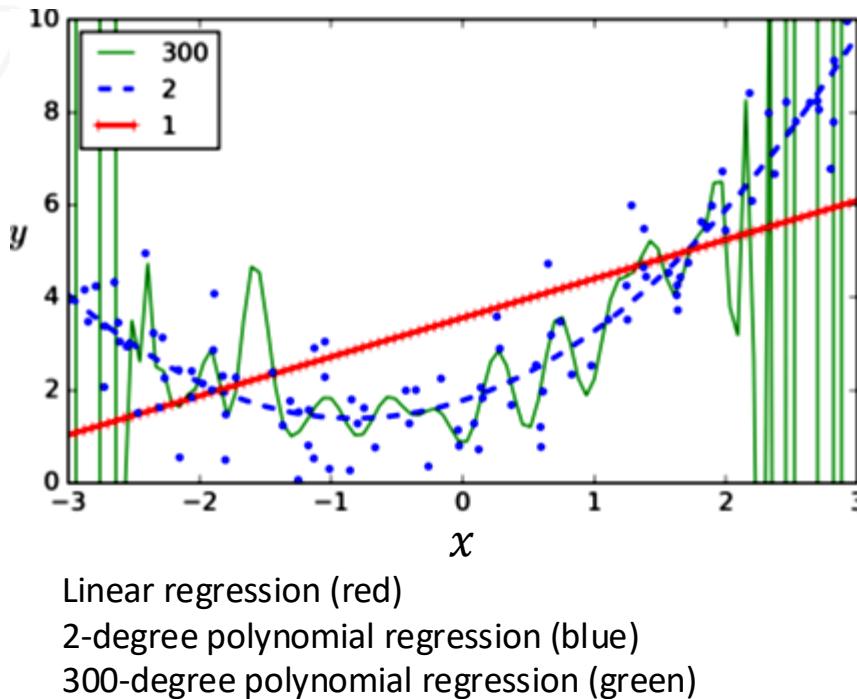
- Helps determine hyperparameters
- More rigorous and randomized than single validation split
- A cross-validation procedure is performed for each combination of hyperparameters
- The general procedure is as follow:
 1. Shuffle the dataset randomly
 2. Split the dataset into k groups
 3. For each group
 - 1) Take that group as a hold-out or validation dataset
 - 2) Combine the remaining $k-1$ groups as one training dataset
 - 3) Fit a model on the training set and evaluate it on the validation (hold-out) set
 - 4) Retain the evaluation scores and discard the model
 4. Average the scores of the model to get a single k -fold validation score



Model Validation

Model Complexity vs Prediction Error

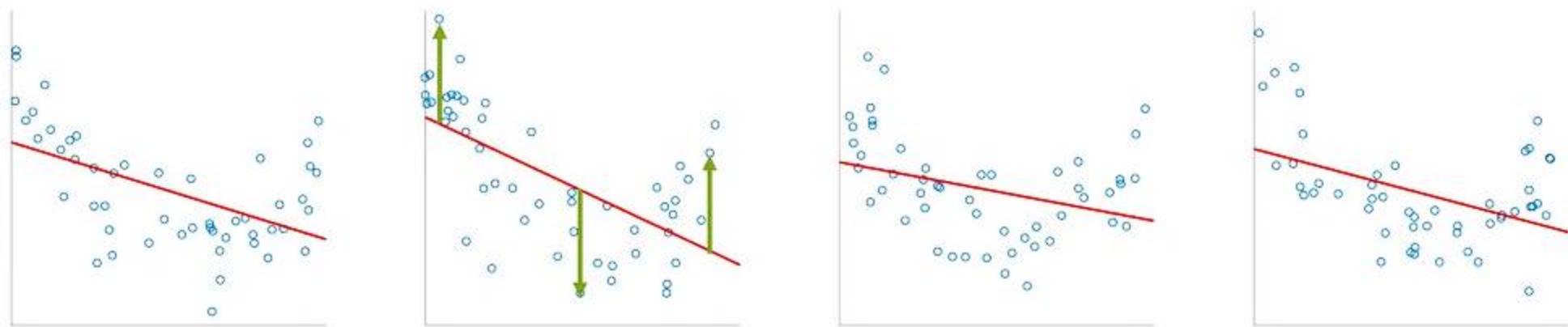
- Model complexity increases with increasing degrees of freedom, e.g., the degree of polynomial regression
- Increasing model complexity:
 - consistently decreases training error at first
 - model starts to overfit to the training data after testing error decreases to a certain complexity level



Model Validation

Bias-Variance Tradeoff

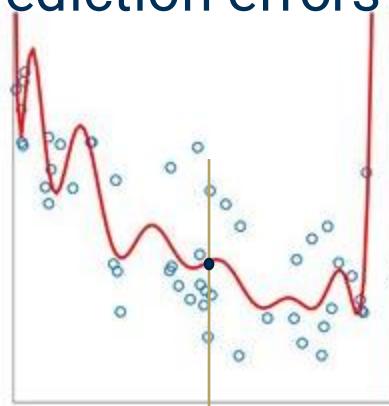
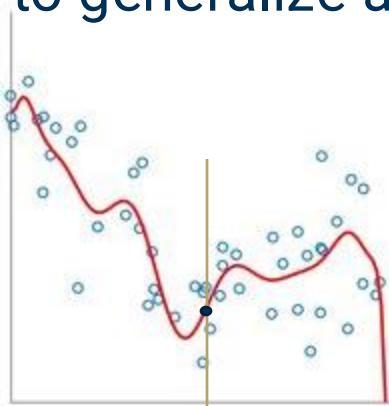
- Assuming you could train a Linear Regression model more than once, each time based on new data.
- The resulting models will have a range of prediction scores.
- Bias is the average prediction scores of these models.
- Models that give high average error are considered to have **high-bias**, and this is directly related to the model's **low complexity**, and lack of flexibility



Model Validation

Bias-Variance Tradeoff

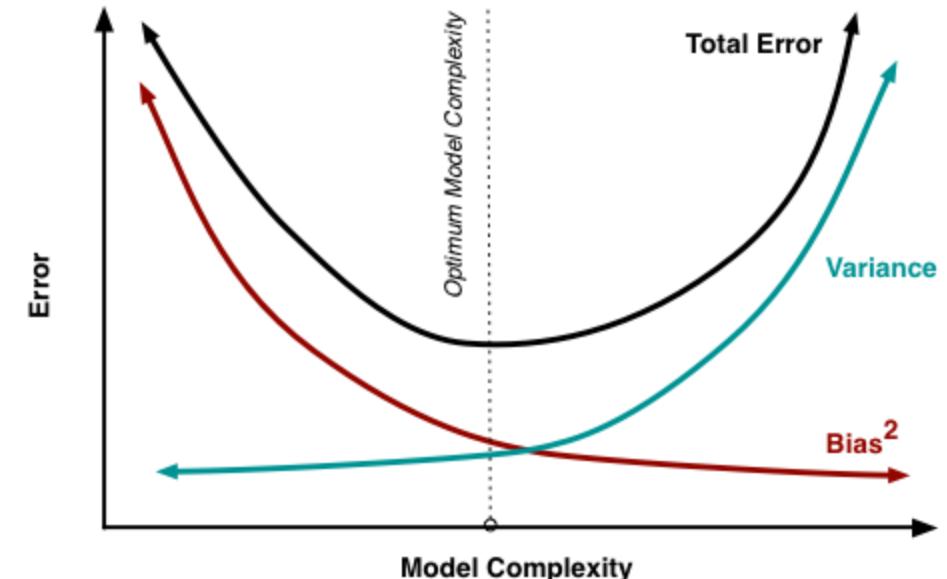
- Assume the same with a high-degree Polynomial Regression model
- Average prediction error (bias) of the resulting polynomial models will be much lower.
- However, the predictions for a given test point will vary largely between the models.
- High-variance occurs when a model is **too flexible** that it **overfits** to the specific data it was trained on.
- High-variance models fail to generalize and therefore give high prediction errors with unseen data



Model Validation

Bias-Variance Tradeoff

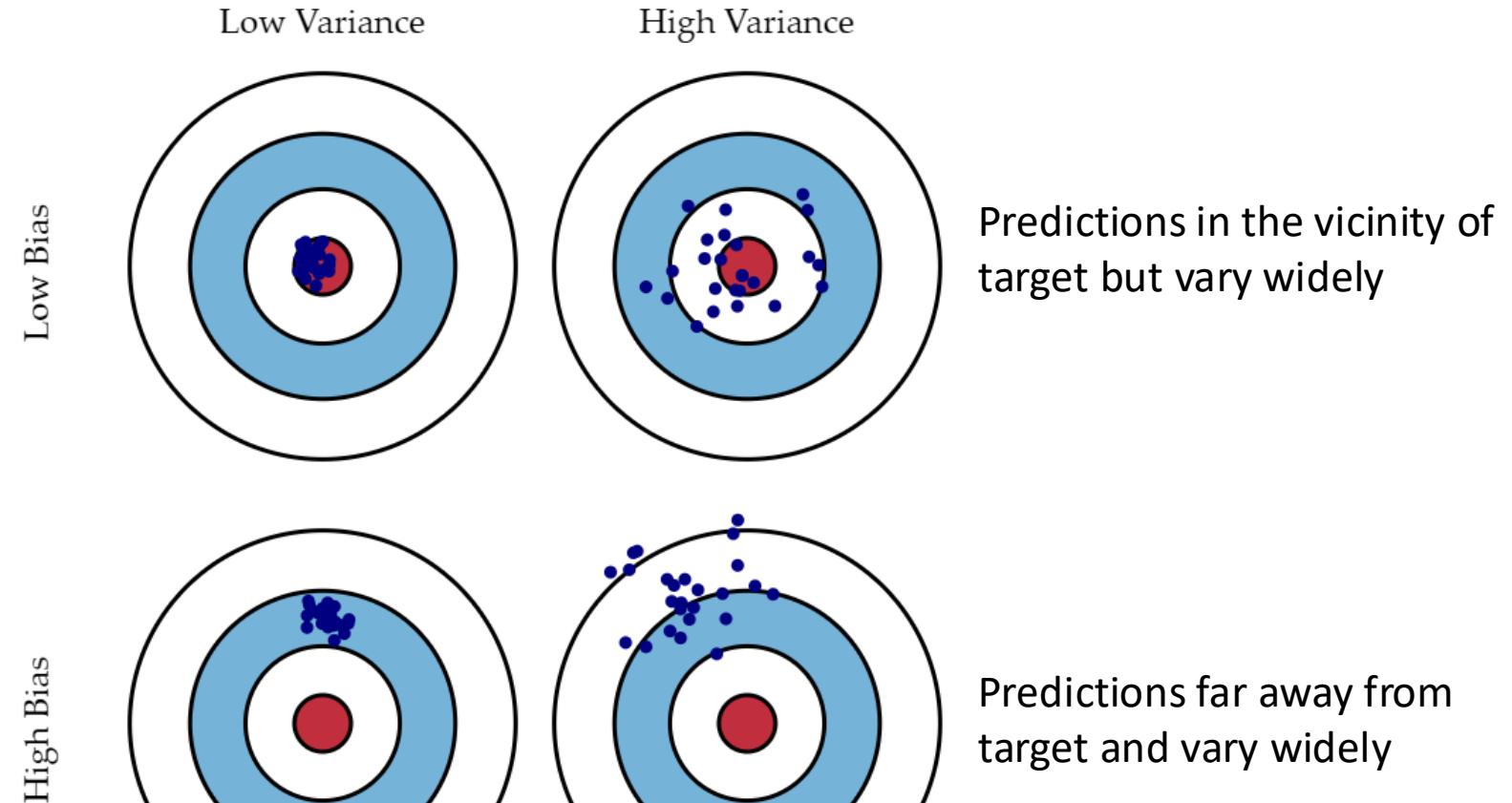
- A model's error score and its failure to generalize is directly related to its complexity (flexibility).
- Optimal model complexity is a tradeoff between Bias and Variance
 - More complexity decreases Bias
 - More complexity increases Variance



Model Validation

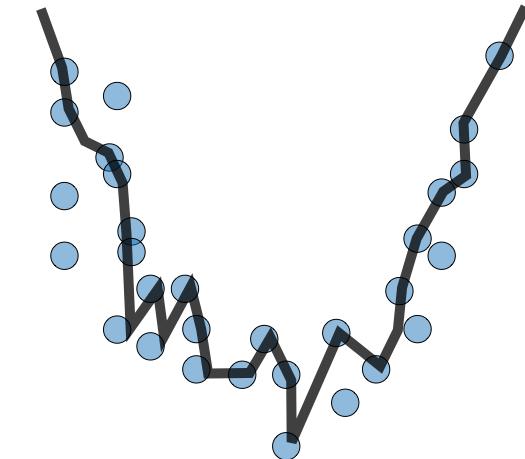
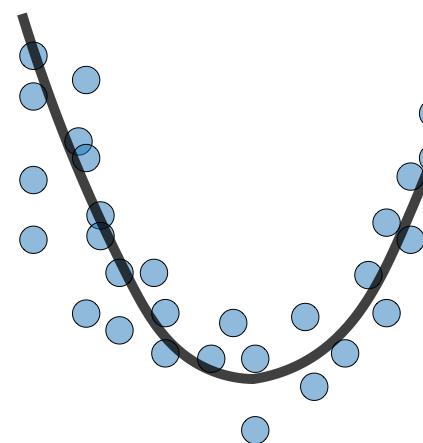
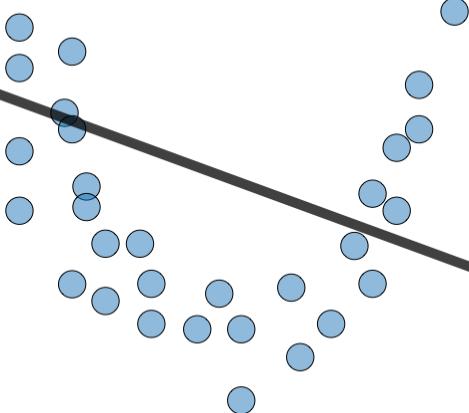
Bias-Variance Tradeoff

predictions close to target
and don't vary much



Model Validation

Bias-Variance Tradeoff



	Linear Model	Low-degree Polynomial	High-degree Polynomial
Bias	VERY HIGH Linear model cannot fit non-linear data. <i>Underfitting</i>	LOW Training error slightly lower than test error	VERY LOW Model is well fixed on training data. <i>Overfitting</i>
Variance	LOW No significant difference in error when measured across different datasets	LOW Model performance on other datasets will not vary significantly.	HIGH Model will show variance performance on other datasets.

Model Validation

Bias-Variance Tradeoff

- Assume x is a test data sample and $f(x)$ its true target
- $\hat{f}(x)$ is a model's prediction of x
- $E[\hat{f}(x)]$ is the average of target predictions given by the trained models
- Bias is the square of the difference between predicted and true target

$$(E[\hat{f}(x)] - f(x))^2$$

- Variance is variance of all predictions made by the different models

$$E[(\hat{f}(x) - E[\hat{f}(x)])^2]$$

- The prediction error made by a model is decomposed as:

$$\text{Error}(x) = E[(f(x) - \hat{f}(x))^2] = \underbrace{(E[\hat{f}(x)] - f(x))^2}_{\text{Bias}^2} + \underbrace{E[(\hat{f}(x) - E[\hat{f}(x)])^2]}_{\text{Variance}} + \sigma_e^2$$

Bias²

Variance

Irreducible
Error

Regression Demo

The following is a visualization of how fitting a set of data points to the best linear or polynomial curve would look in 2D.

Data:

X	Y
-4	1
-2	5
5	2
6	4
13	12

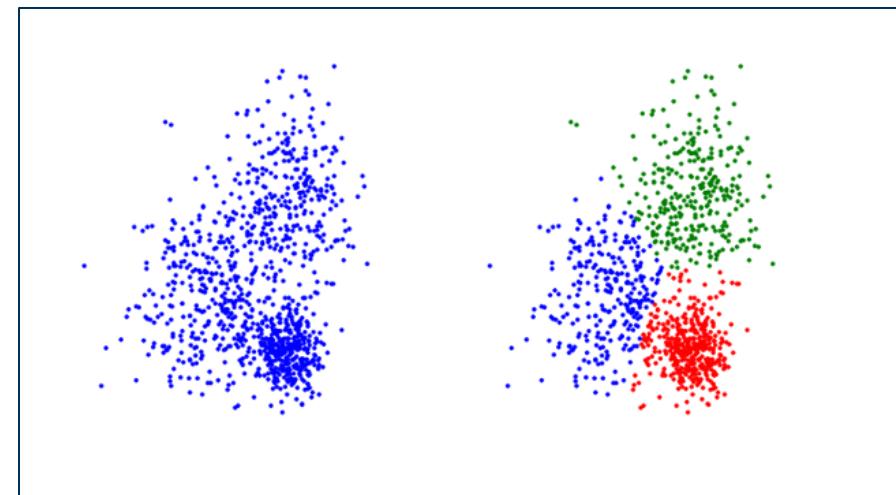
[Linear Regression Visualization](#)

[Polynomial Regression Visualization](#)

ECE 4803/8803: Fundamentals of Machine Learning (FunML)

Spring 2024

Lecture 10 Part 2: Clustering



Overview

In This Lecture..

Introduction

- What is clustering?
- Clustering application areas
- Simple example
- Definitions
- What we need for clustering
- Formalization of the clustering problem
- Proximity measures

Common Clustering Methods

Evaluating Clustering Performance

Image Segmentation as Clustering Problem

Introduction

Types of Learning

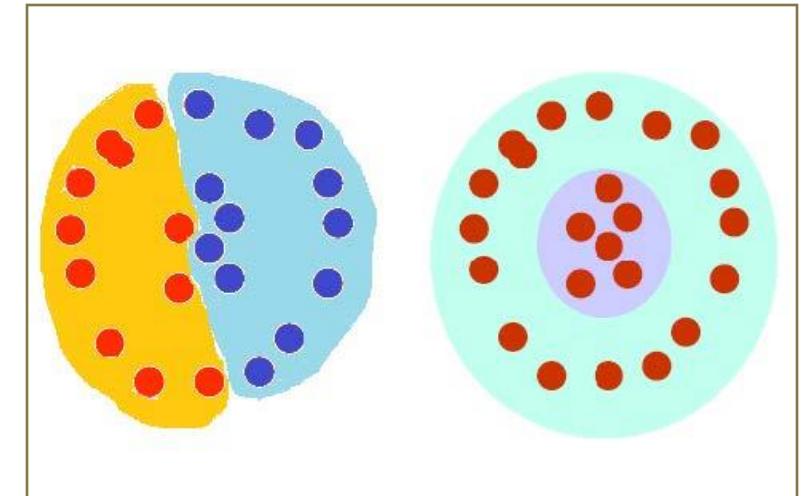
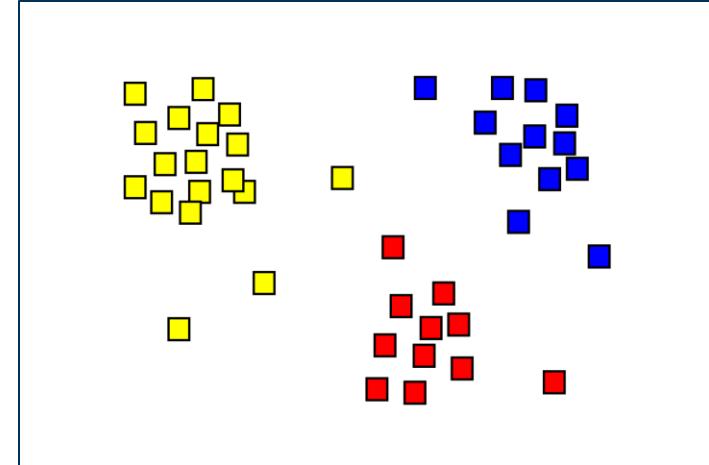
- **Supervised Learning**
 - Requires labeled training data $(x_1, y_1), \dots, (x_N, y_N)$
 - Learns by sample-label pairs
 - e.g., Images of cats (with label 0) and dogs (with label 1)
- **Unsupervised Learning**
 - No labels required x_1, \dots, x_N
 - Learns to group samples of similar characteristics
 - e.g., Given a set of audio clips, group them according to the speaker's voice.

Introduction

What is Clustering?

Clustering is the **unsupervised** task of grouping a set of samples in such a way where samples in the same group (called a **cluster**) are *more similar* (in some sense) to each other than to those in other groups (clusters).

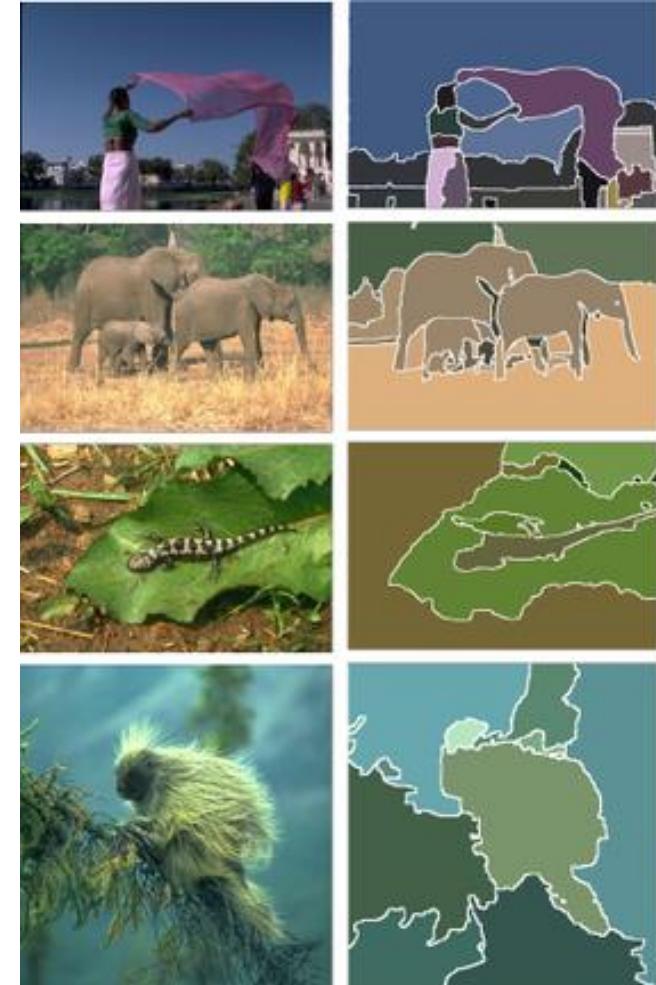
- This implies a need for:
 - Specific features to compare samples
 - Proximity measures to assess the similarity of the samples
 - Clustering algorithms to compute proximity and assigns same labels to similar samples
- Unlike classification or regression, clustering is an unsupervised learning task since data is unlabeled (no given target labels).



Clustering

Image Segmentation

- The process of partitioning a digital image into multiple segments (regions).
- The result of image segmentation is a set of segments that collectively cover the entire image, or as a set of contours of such regions.
- Each of the pixels in a region are similar with respect to some characteristic or computed property (feature) such as color, intensity, or texture.



Clustering

Grouping News Articles

Clustering news articles

- News articles dynamically listed in groups and subgroups
- Each article is a data instance
- Words in articles constitute features based on which articles are clustered

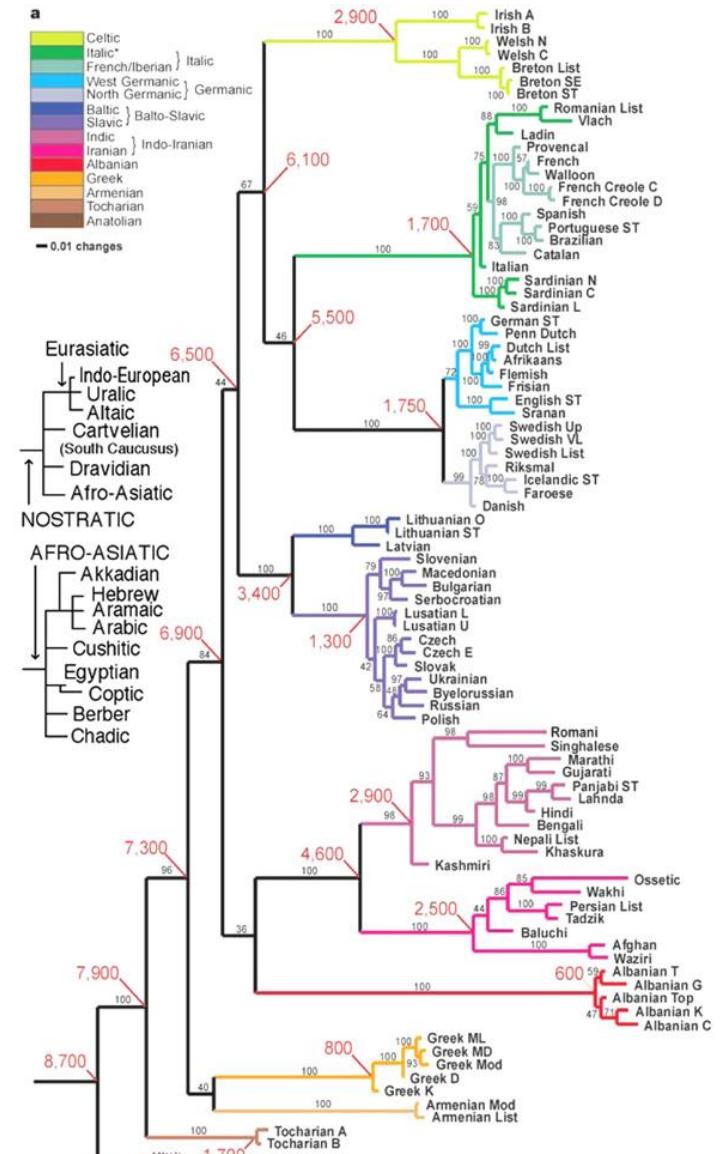
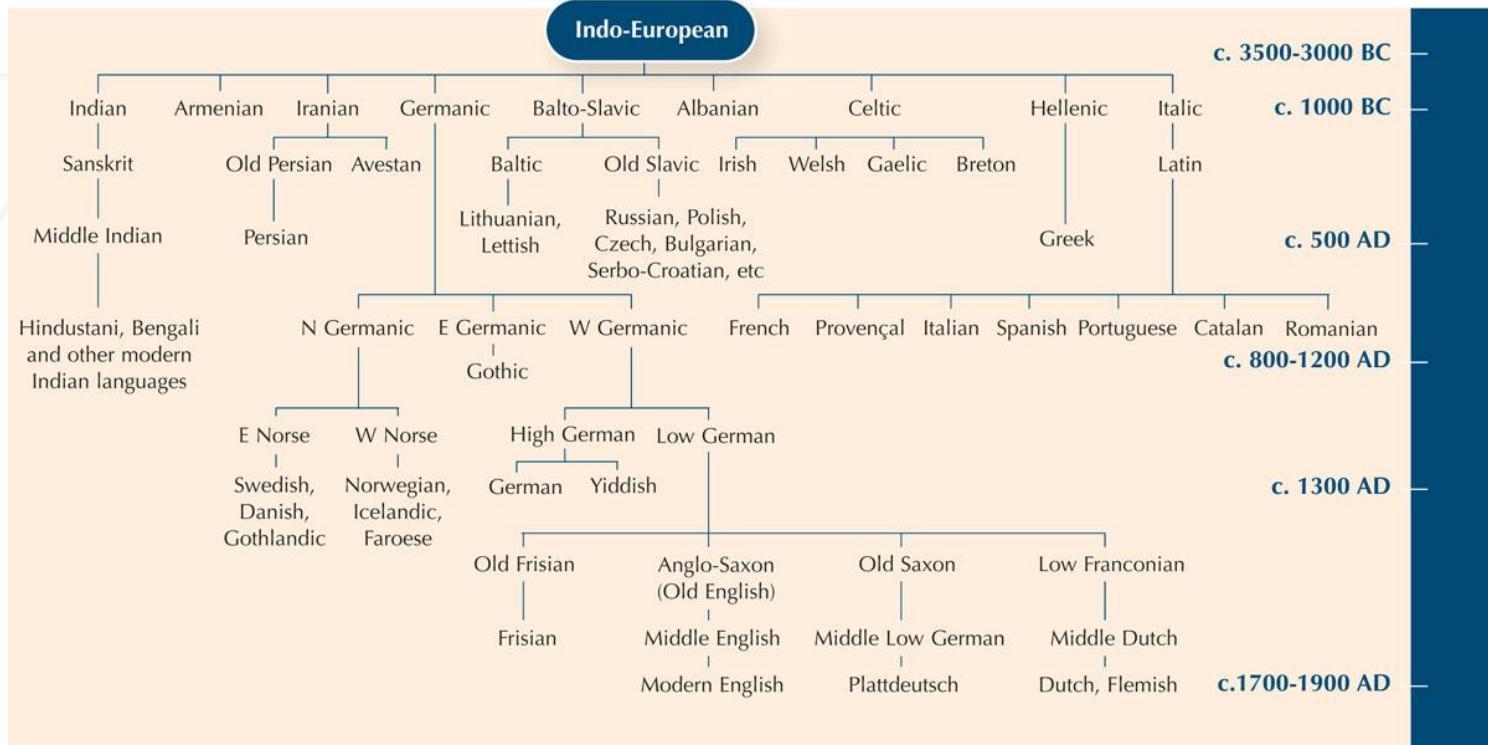
The screenshot shows the Google News homepage. On the left, a sidebar lists main groupings: Top stories, For you, Favorites, Saved searches, U.S., World, Local, Business (which is selected and highlighted in blue), Technology, Entertainment, Sports, Science, and Health. Below these are Language & region (English | United States) and Settings. A large bracket on the right side of the sidebar is labeled "Main groupings". The main content area is titled "Business" and shows a sub-grouping under "Business group". It includes tabs for Latest, Economy, Markets (which is selected and highlighted in blue), Jobs, Personal finance, and Entrepreneurship. Four news articles are listed:

- Dow Jones Futures: Current Stock Market Rally – 4 Reasons To Be Bullish (Investor's Business Daily • 4 hours ago)
- Asia stocks gain as China's economic activity unexpectedly bounces in March (CNBC • 14 minutes ago)
- 'New Dow' separates from DowDuPont April 1, to trade on stock market as DOW (MLive.com • 14 hours ago)
- Next recession: A signal less obvious than yield curve already here (Business Insider • 16 hours ago)

Each article has a small thumbnail image to its right.

Clustering Languages

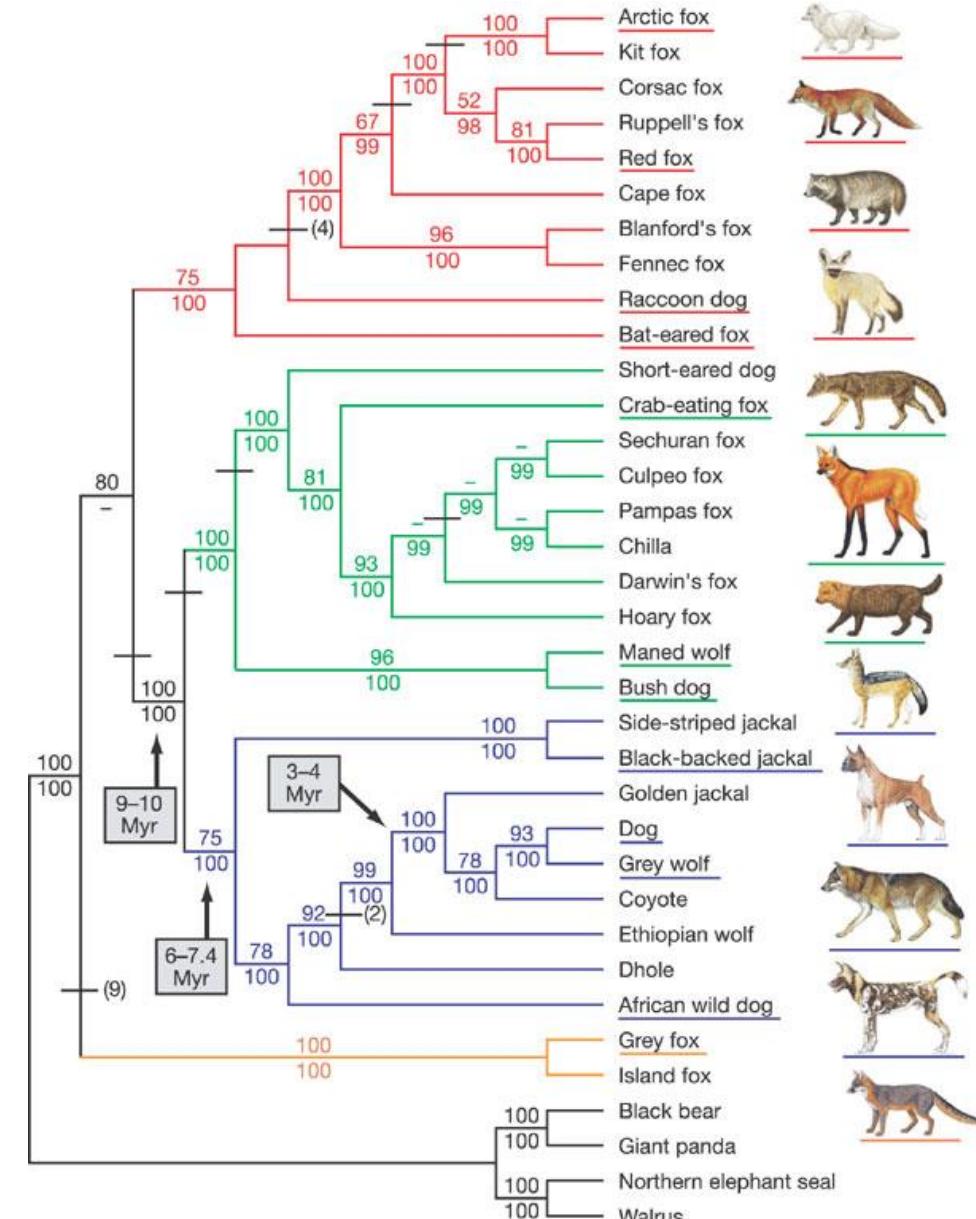
Clustering languages



Clustering Species

Clustering species

- Clustering individual animals by species based upon various characteristics and behaviors
- An animal is represented by a set of features (characteristics or behaviors)



Clustering

Applications

- Image segmentation
- Clustering gene expression data
- Clustering news articles
- Clustering languages
- Clustering species
- Recommender systems
- Human genetic clustering
- Genome sequence analysis
- Analysis of antimicrobial activity
- Grouping of shopping items
- Search result grouping
- Slippy map optimization
- Crime analysis
- Climatology
- etc.

Clustering

Clustering via ChatGPT

- The following phrases are all taken from two different sources. Can you cluster them into two groups based on their content:
- "The Seven Social Sins are: Wealth without work. Pleasure without conscience. Knowledge without character. Commerce without morality. Science without humanity. Worship without sacrifice. Politics without principle."
- "The death toll from a magnitude 7.8 earthquake and its aftershocks, which struck the southeastern region of Turkey along the border with Syria, has continued to rise."
- "Bucharest has denied that Russian missiles flew over Romania, countering Kyiv's claims that two of Moscow's rockets crossed the NATO member's airspace on their way to Ukraine."
- "Reader, suppose you were an idiot. And suppose you were a member of Congress. But I repeat myself."

CHATGPT

The phrases can be grouped as follows:

Group 1:

"The Seven Social Sins are:
Wealth without work.
Pleasure without conscience.
Knowledge without character.
Commerce without morality.
Science without humanity.
Worship without sacrifice.
Politics without principle."

"Reader, suppose you were an idiot. And suppose you were a member of Congress. But I repeat myself."

Group 2:

"The death toll from a magnitude 7.8 earthquake and its aftershocks, which struck the southeastern region of Turkey along the border with Syria, has continued to rise."

"Bucharest has denied that Russian missiles flew over Romania, countering Kyiv's claims that two of Moscow's rockets crossed the NATO member's airspace on their way to Ukraine."

Clustering

Clustering via ChatGPT

Can you explain to me what criteria did you use to perform this grouping

ChatGPT:

The criteria I used to perform this grouping was based on the content and tone of the phrases. For Group 1, both phrases are more philosophical in nature and contain reflections on social and political issues. For Group 2, both phrases are factual and present information about current events and incidents. This distinction between philosophical and factual content was the main criteria I used to group these phrases into two categories.

Clustering

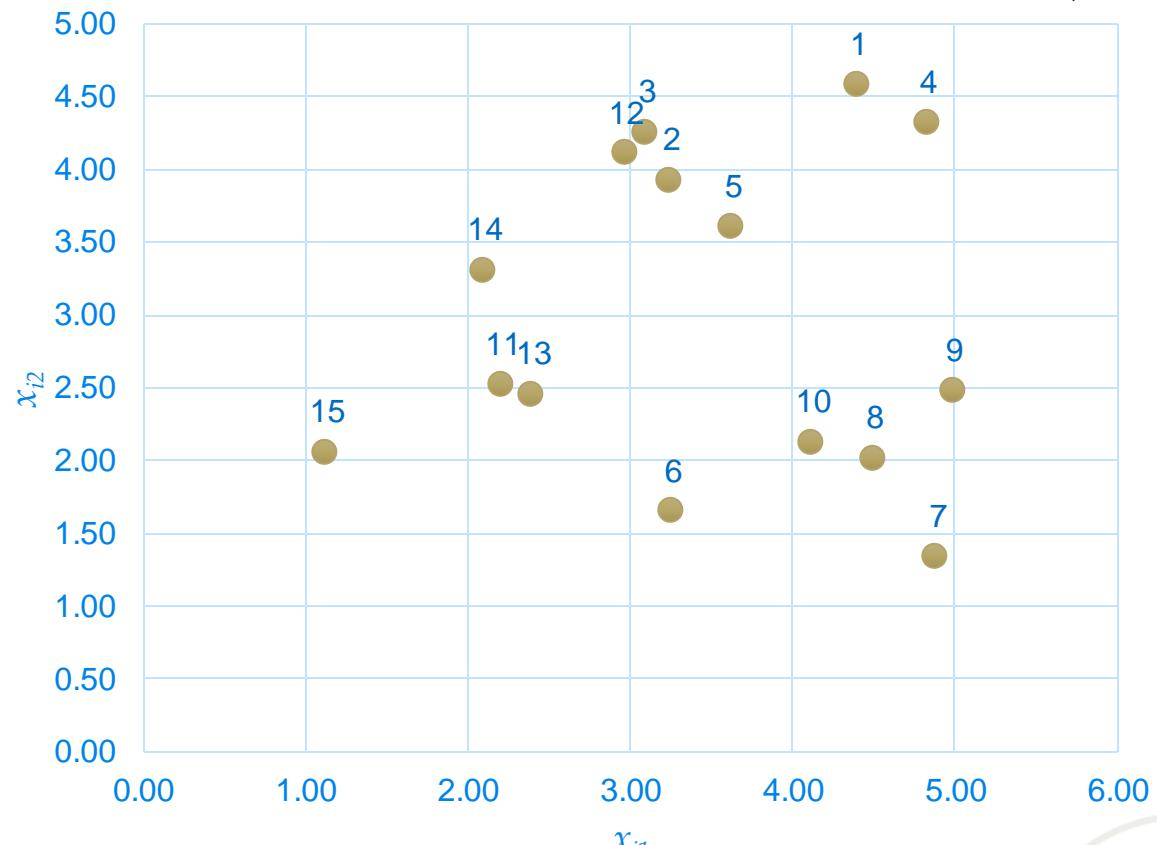
A Simple Example

Suppose that we are given 15 **unlabeled** samples x_1, \dots, x_{15} in 2-dimensional feature space

2 features

Sample	x_{i1}	x_{i2}
1	4.40	4.57
2	3.25	3.92
3	3.10	4.25
4	4.83	4.31
5	3.63	3.60
6	3.26	1.64
7	4.89	1.33
8	4.50	2.01
9	4.99	2.47
10	4.12	2.12
11	2.21	2.51
12	2.97	4.10
13	2.40	2.45
14	2.10	3.30
15	1.13	2.05

15 data samples



Data matrix:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \\ \vdots & \vdots \\ x_{15,1} & x_{15,2} \end{bmatrix}$$

Clustering

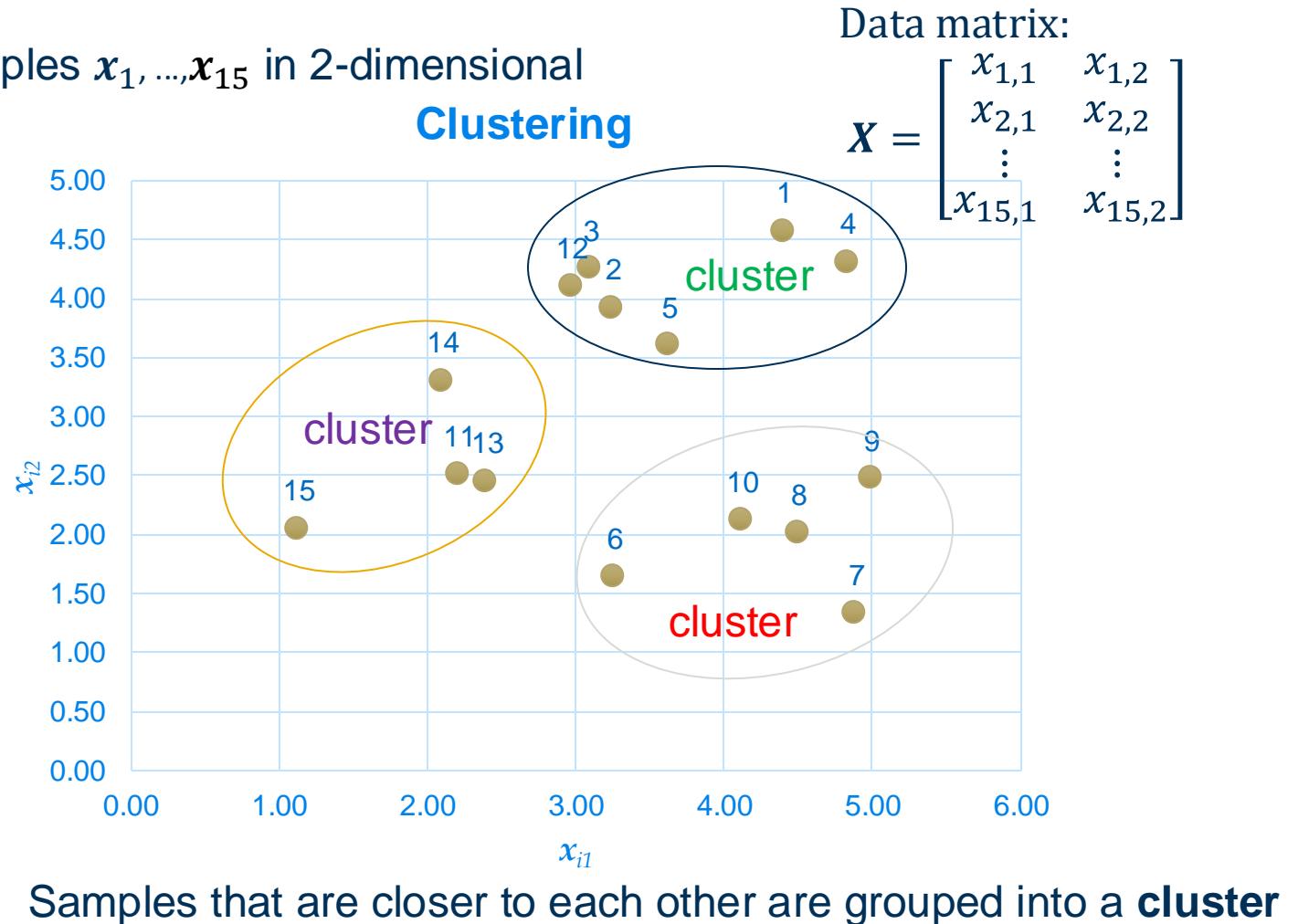
A Simple Example

Suppose that we are given 15 **unlabeled** samples x_1, \dots, x_{15} in 2-dimensional feature space

Sample	x_{i1}	x_{i2}
1	4.40	4.57
2	3.25	3.92
3	3.10	4.25
4	4.83	4.31
5	3.63	3.60
6	3.26	1.64
7	4.89	1.33
8	4.50	2.01
9	4.99	2.47
10	4.12	2.12
11	2.21	2.51
12	2.97	4.10
13	2.40	2.45
14	2.10	3.30
15	1.13	2.05

Data matrix:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \\ \vdots & \vdots \\ x_{15,1} & x_{15,2} \end{bmatrix}$$



Clustering

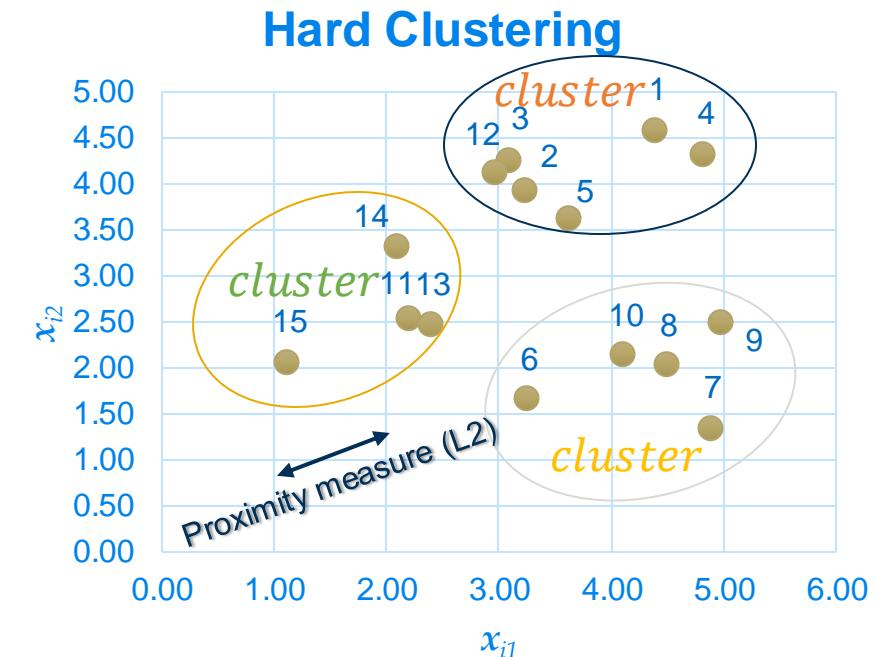
Terminologies

- **Features:** a sample x_i is represented as a set of features $\{x_{i1}, \dots, x_{iP}\}$ in the P -dimensional feature space, x_{ij} is a single feature.
- **Proximity measure:** for two data samples x_j and x_k , a proximity measure indicates how similar or dissimilar the two samples are
- **Cluster:** a group of similar data samples.
- **Hard Clustering (crisp clustering):** each sample is assigned to one and only one cluster.
- **Fuzzy Clustering (soft clustering):** each sample can belong to more than one cluster with some degree of membership.

2 features

15 data samples

Sample	x_{i1}	x_{i2}
1	4.40	4.57
2	3.25	3.92
3	3.10	4.25
4	4.83	4.31
5	3.63	3.60
6	3.26	1.64
7	4.89	1.33
8	4.50	2.01
9	4.99	2.47
10	4.12	2.12
11	2.21	2.51
12	2.97	4.10
13	2.40	2.45
14	2.10	3.30
15	1.13	2.05



Clustering

Formalization of the Clustering Problem

- Given a dataset $X = \{x_1, x_2, \dots, x_N\}$ where $x_i = [x_{i1}, \dots, x_{ij}, \dots, x_{iP}]^T$ is a sample in the P -dimensional feature space, N is the number of samples
- The clustering of X is the partitioning of X into K clusters $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K\}$ satisfying the following conditions:
 - Each sample should be assigned to a cluster, i.e.

$$\bigcup_{k=1}^K \mathcal{C}_k = X$$

- Each cluster has at least one sample assigned to it, i.e.

$$\mathcal{C}_k \neq \emptyset, k = 1, \dots, K$$

- For hard clustering, each sample is assigned to one and only one cluster, i.e.

$$\mathcal{C}_k \bigcap \mathcal{C}_j = \emptyset, \quad \text{where } k \neq j$$

Clustering

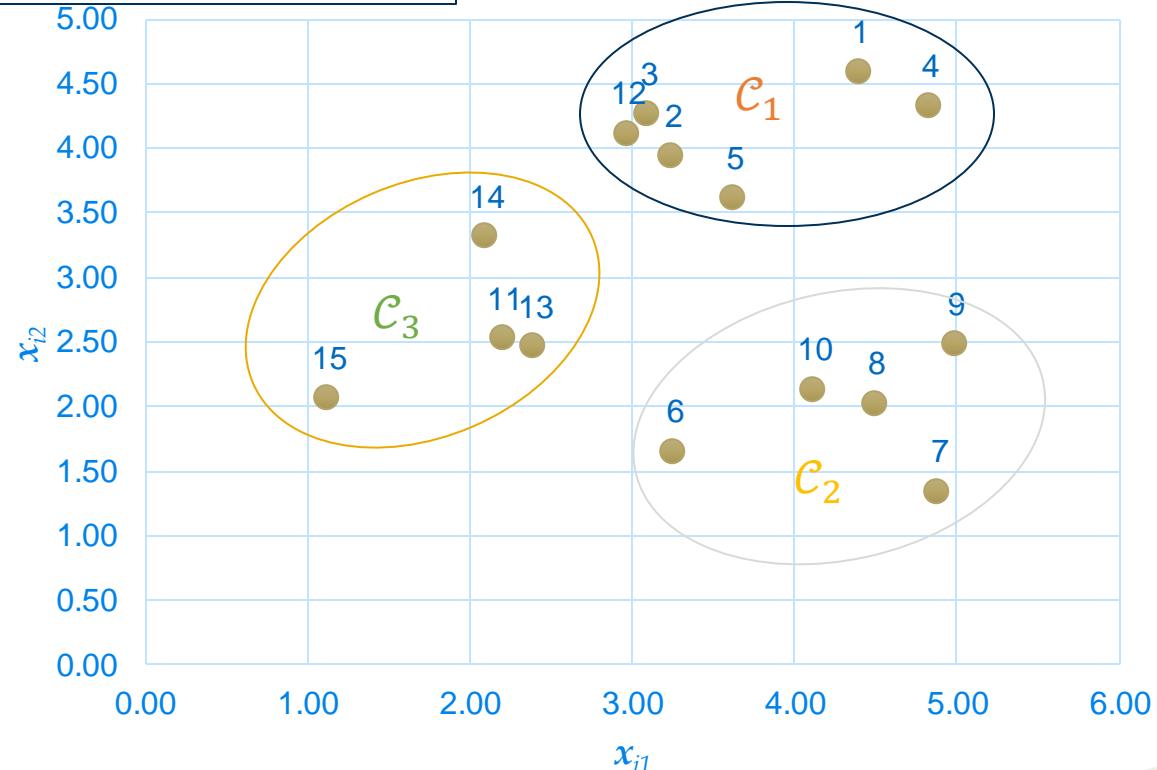
Formalization of the Clustering Problem

Sample	x_{i1}	x_{i2}
1	4.40	4.57
2	3.25	3.92
3	3.10	4.25
4	4.83	4.31
5	3.63	3.60
6	3.26	1.64
7	4.89	1.33
8	4.50	2.01
9	4.99	2.47
10	4.12	2.12
11	2.21	2.51
12	2.97	4.10
13	2.40	2.45
14	2.10	3.30
15	1.13	2.05

2 features

- $\bigcup_{k=1}^3 \mathcal{C}_k = \mathcal{X}$
- $\mathcal{C}_k \neq \emptyset, k = 1, 2, 3$
- $\mathcal{C}_k \cap \mathcal{C}_j = \emptyset$ for $k \neq j$

Clustering



Data matrix:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \\ \vdots & \vdots \\ x_{15,1} & x_{15,2} \end{bmatrix}$$

Clustering

Proximity Measures

- Similarity vs. Dissimilarity
- Cosine Similarity
- Euclidean Distance
- Manhattan Distance
- Minkowski Distance
- Mahalanobis Distance

Proximity Measures

Similarity vs Dissimilarity

For two data samples x_j and x_k , proximity is measured in terms of either:

- Similarity function, $s(x_j, x_k)$ which has a large value if x_j, x_k are similar (ranges between two limits such as from 0 for least similar to 1 for most similar)
- Dissimilarity (or distance) function, $d(x_j, x_k)$ which has a small value if x_j, x_k are similar (ranges from 0 to $+\infty$)

Proximity Measures

Cosine Similarity

- A **similarity** measure between feature vectors of an inner product space

$$s(x_j, x_k) = \cos(\theta) = \frac{x_j^T x_k}{\|x_j\| \|x_k\|} = \frac{\sum_{i=1}^P x_{ji} x_{ki}}{\sqrt{\sum_{i=1}^P x_{ji}^2} \sqrt{\sum_{i=1}^P x_{ki}^2}}$$

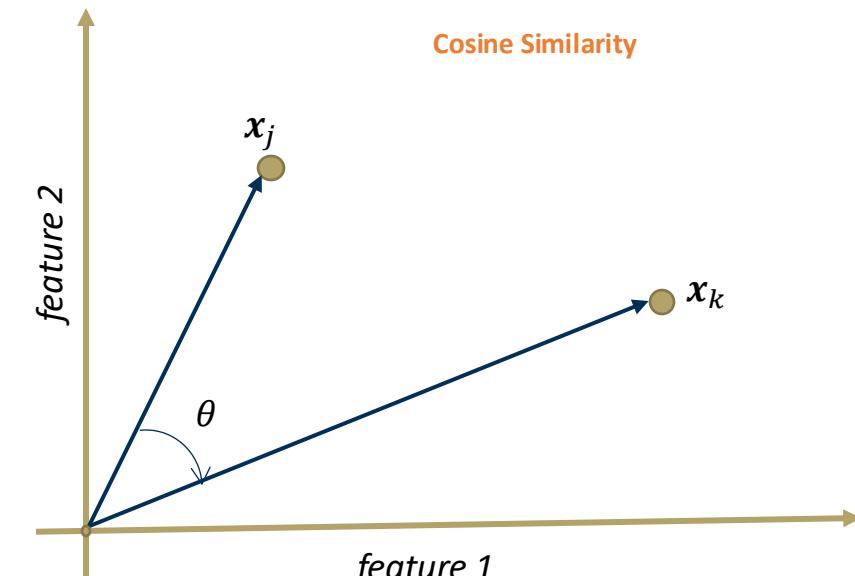
independent of the magnitude

where θ is the **angle** between the two feature vectors x_j and x_k

- $\cos(\theta) \in [-1, 1]$, where:
 - $\cos(\theta) = 1$: maximum similarity ($\theta=0$, features with the **same direction**)
 - $\cos(\theta) = -1$: minimum similarity ($\theta=\pi$, features with the **opposite direction**)
 - $\cos(\theta) = 0$: uncorrelated ($\theta=\frac{\pi}{2}$, features with the **orthogonal direction**)
- cosine distance: $d = 1 - \cos(\theta)$, $d \in [0, 2]$

inner product of x_j, x_k :
 $\langle x_j, x_k \rangle = x_j^T x_k$

Cosine Similarity



Proximity Measures

Cosine Similarity

- Example 1:

Given the feature vectors $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{10}$: $\mathbf{x}_1^T \mathbf{x}_2 = \sum_{i=1}^{10} x_{1i} x_{2i} = 5$

$$\mathbf{x}_1 = \begin{bmatrix} 3 \\ 2 \\ 0 \\ 5 \\ 0 \\ 0 \\ 0 \\ 2 \\ 0 \\ 0 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 2 \end{bmatrix}$$

Sparse vectors with many 0s entries

$$\|\mathbf{x}_1\| = \sqrt{\sum_{i=1}^{10} x_{1i}^2} = \sqrt{42} = 6.481$$

$$\|\mathbf{x}_2\| = \sqrt{\sum_{i=1}^{10} x_{2i}^2} = \sqrt{6} = 2.245$$

$$\text{Cosine similarity } s(\mathbf{x}_1, \mathbf{x}_2) = \frac{5}{6.481 \times 2.245} = 0.315$$

$$\text{Cosine distance } d(\mathbf{x}_1, \mathbf{x}_2) = 1 - s(\mathbf{x}_1, \mathbf{x}_2) = 0.685$$

Cosine similarity is very efficient to evaluate, especially for sparse vectors with many 0s entries.

Proximity Measures

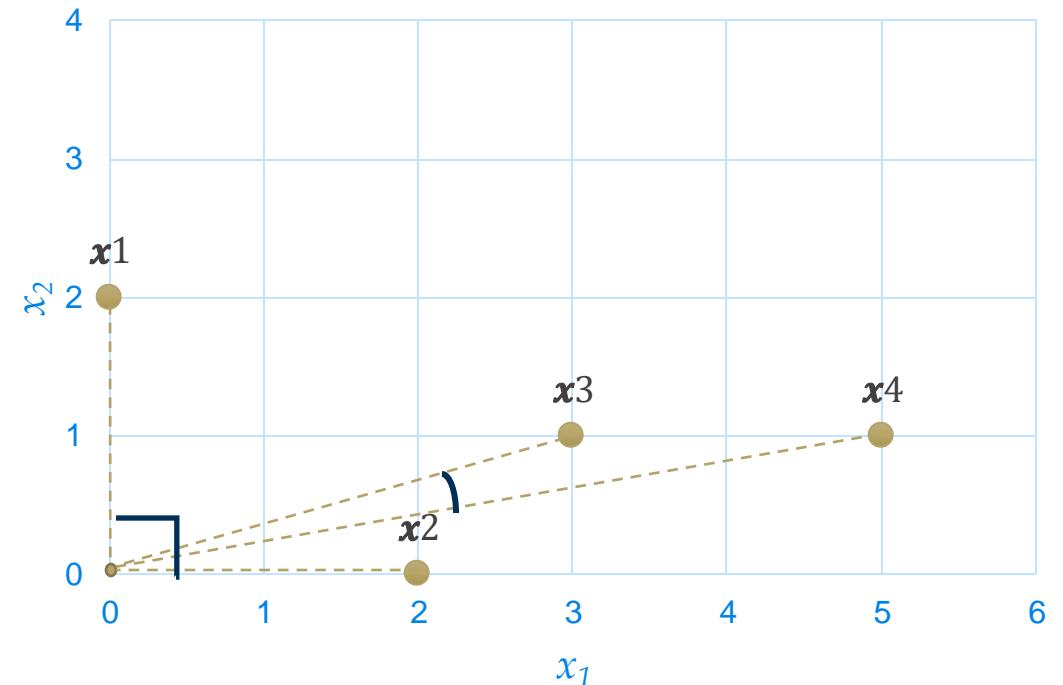
Cosine Similarity

Example 2: Given the samples $x_1, x_2, x_3, x_4 \in \mathbb{R}^2$

samples	$x_{i,1}$	$x_{i,2}$
x_1	0	2
x_2	2	0
x_3	3	1
x_4	5	1

Pairwise cosine similarity

$s(x_1, x_2)$	x_1	x_2	x_3	x_4
x_1	1	0	0.3162	0.1961
x_2	0	1	0.9487	0.9806
x_3	0.3162	0.9487	1	0.9923
x_4	0.1961	0.9806	0.9923	1



x_1, x_2 are least similar with orthogonal direction
 x_3, x_4 are most similar with smallest angle

Proximity Measures

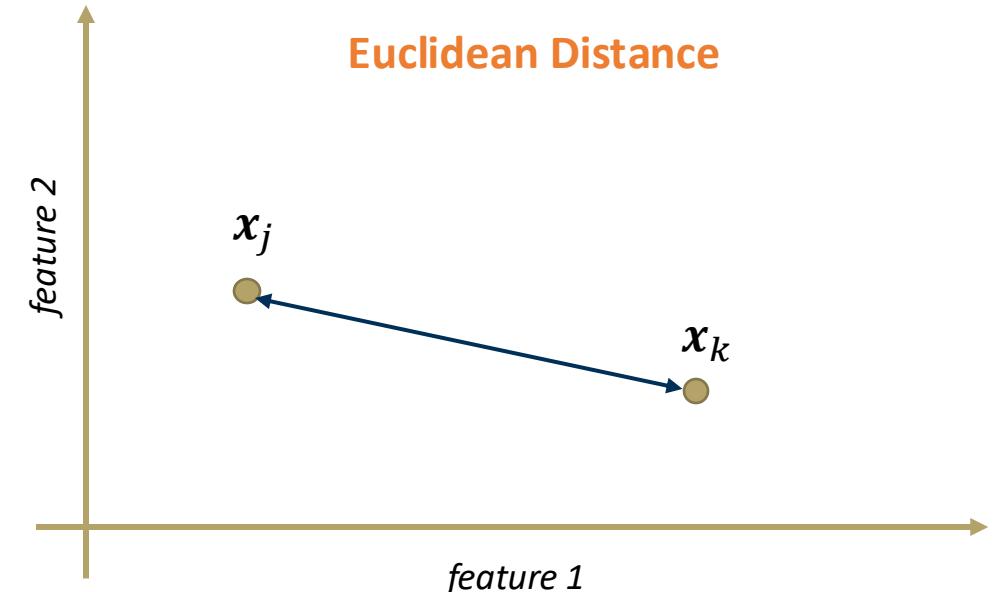
Euclidean Distance

- A **distance measure** in Euclidean space

$$d(x_j, x_k) = \sqrt{\sum_{i=1}^P (x_{ji} - x_{ki})^2}$$

where P is the number of dimensions, and x_{ji} and x_{ki} are, respectively, the i^{th} features of samples x_j and x_k .

- $d(x_j, x_k) \in [0, +\infty]$, where:
 - 0 is maximum similarity
 - $+\infty$ is minimum similarity
- The Euclidean distance between two points is the length of the path connecting them according to Pythagorean theorem.
- Euclidean distance is the most used distance measure.
- Most suitable when data are dense or non-categorical.



Proximity Measures

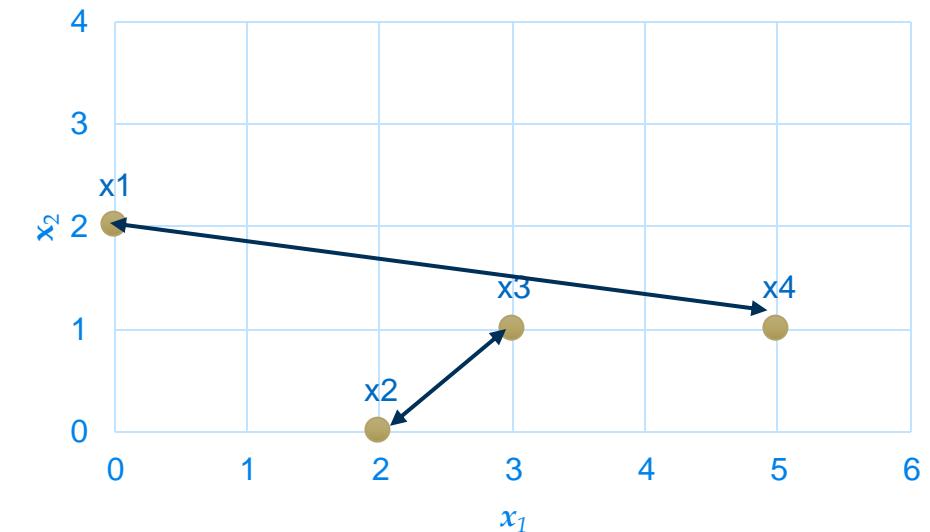
Euclidean Distance

- Example: Given the same samples $x_1, x_2, x_3, x_4 \in \mathbb{R}^2$

samples	$x_{i,1}$	$x_{i,2}$
x_1	0	2
x_2	2	0
x_3	3	1
x_4	5	1

Pairwise Euclidean distance

$d(x_i, x_j)$	x_1	x_2	x_3	x_4
x_1	0	2.828	3.162	5.099
x_2	2.828	0	1.414	3.162
x_3	3.162	1.414	0	2
x_4	5.099	3.162	2	0



x_1, x_4 are least similar with largest distance

x_2, x_3 are most similar with smallest distance

Proximity Measures

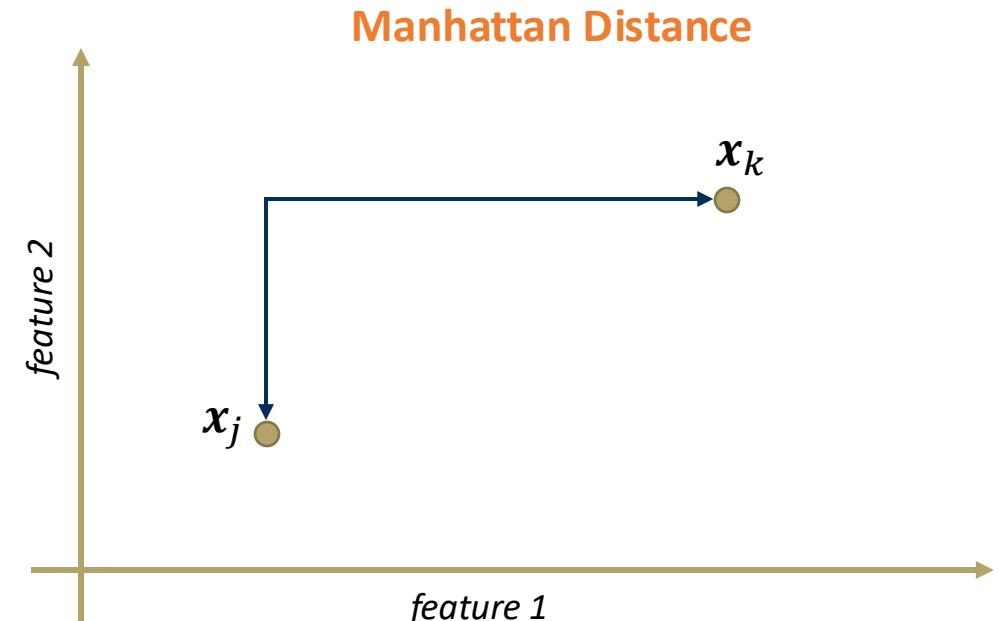
Manhattan Distance

- A **distance** measure calculating the sum of the *absolute differences* of corresponding features

$$d(x_j, x_k) = \sum_{i=1}^P |x_{ji} - x_{ki}|$$

where P is the number of dimensions, and x_{ji} and x_{ki} are, respectively, the i^{th} features of samples x_j and x_k .

- $d(x_j, x_k) \in [0, +\infty]$, where:
 - 0 is maximum similarity
 - $+\infty$ is minimum similarity
- Also known as Manhattan length, rectilinear distance, L1 distance or L1 norm, city block distance.
- Manhattan distance is preferable to Euclidean distance for high dimensional and sparse data



Proximity Measures

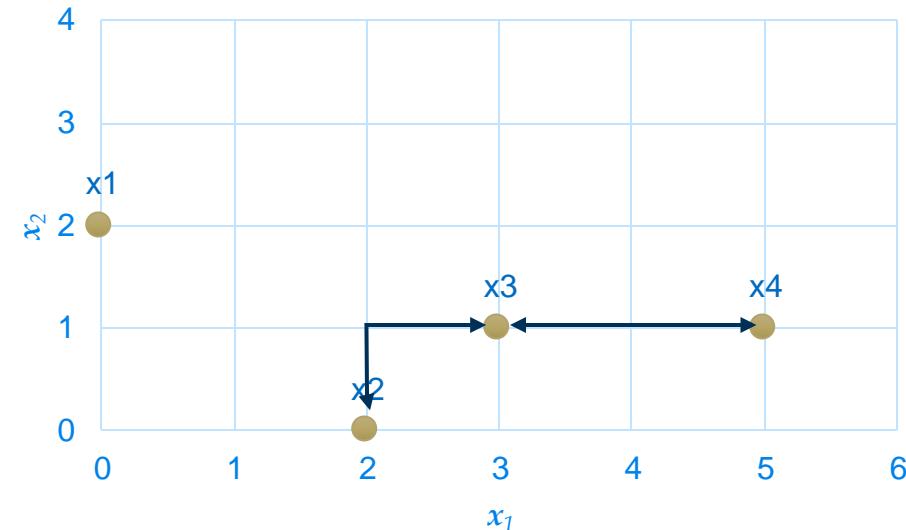
Manhattan Distance

- Example: Given the same samples $x_1, x_2, x_3, x_4 \in \mathbb{R}^2$

samples	$x_{i,1}$	$x_{i,2}$
x_1	0	2
x_2	2	0
x_3	3	1
x_4	5	1

Pairwise Manhattan distance

$d(x_1, x_2)$	x_1	x_2	x_3	x_4
x_1	0	4	4	6
x_2	4	0	2	4
x_3	4	2	0	2
x_4	6	4	2	0



$d(x_3, x_2) = d(x_3, x_4)$ because Manhattan distance is measured by horizontal and vertical proximity

Proximity Measures

Minkowski Distance

- A **distance metric** in a **normed vector space** as a generalization of both the Euclidean distance and the Manhattan distance

$$d(x_j, x_k) = \left(\sum_{i=1}^P |x_{ji} - x_{ki}|^r \right)^{\frac{1}{r}},$$

where r is a parameter, P is the number of dimensions, and x_{ji} and x_{ki} are, respectively, the i^{th} features of samples x_j and x_k .

$r = 1$: Manhattan distance (L_1 norm)

$r = 2$: Euclidean distance (L_2 norm)

$r = \infty$: The maximum element in the difference vector $x_j - x_k$ ("supremum", L_{\max} norm, L_∞ norm)

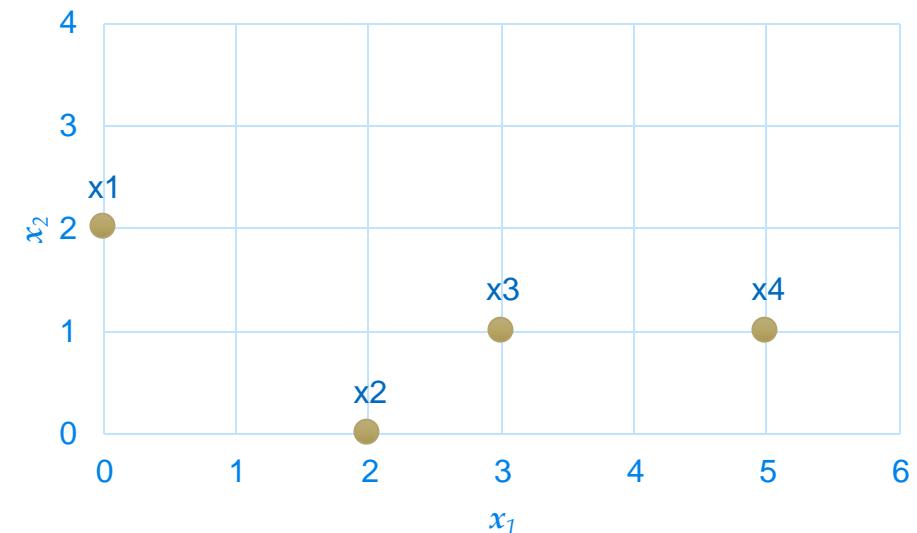
$r = 0$: The number of non-zero elements in the difference vector $x_j - x_k$ (L_0 norm)

Proximity Measures

Minkowski Distance

- Example: Given the same samples $x_1, x_2, x_3, x_4 \in \mathbb{R}^2$

samples	$x_{i,1}$	$x_{i,2}$
x_1	0	2
x_2	2	0
x_3	3	1
x_4	5	1



Pairwise Manhattan distance

$r = 1$	x_1	x_2	x_3	x_4
x_1	0	4	4	6
x_2	4	0	2	4
x_3	4	2	0	2
x_4	6	4	2	0

Pairwise Euclidean distance

$r = 2$	x_1	x_2	x_3	x_4
x_1	0	2.828	3.162	5.099
x_2	2.828	0	1.414	3.162
x_3	3.162	1.414	0	2
x_4	5.099	3.162	2	0

Pairwise L_∞ norm

$r = \infty$	x_1	x_2	x_3	x_4
x_1	0	2	3	5
x_2	2	0	1	3
x_3	3	1	0	2
x_4	5	3	2	0

Pairwise L_0 norm

$r = 0$	x_1	x_2	x_3	x_4
x_1	0	2	2	2
x_2	2	0	2	2
x_3	2	2	0	1
x_4	2	2	1	0

Proximity Measures

Mahalanobis Distance

- A **distance measure** between two random variables x_j, x_k of the **same distribution**, defined as:

$$d(x_j, x_k) = (x_j - x_k)^T \Sigma^{-1} (x_j - x_k)$$

where Σ is the covariance matrix of the **entire dataset X** .

- Useful for **detecting outliers**
- When Σ is an identity matrix, the Mahalanobis distance is the same as the Euclidian distance.

Proximity Measures

Mahalanobis Distance

- **Variance** measures the variation of a single random variable (like the grade of a student in a university)

- $\sigma_x^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$, where N is the number of samples and \bar{x} is the mean of the random variable x

- **Covariance** is a measure of how much two random variables vary together (like the grade of a student and the major of a student in a university)

- $\sigma(x, y) = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$, where N is the number of samples, \bar{x} is the mean of the random variable x , and \bar{y} is the mean of the random variable y

- $\sigma(x, x) = \sigma_x^2$

- For a 2X2 random variables, we have a 2X2 covariance matrix:

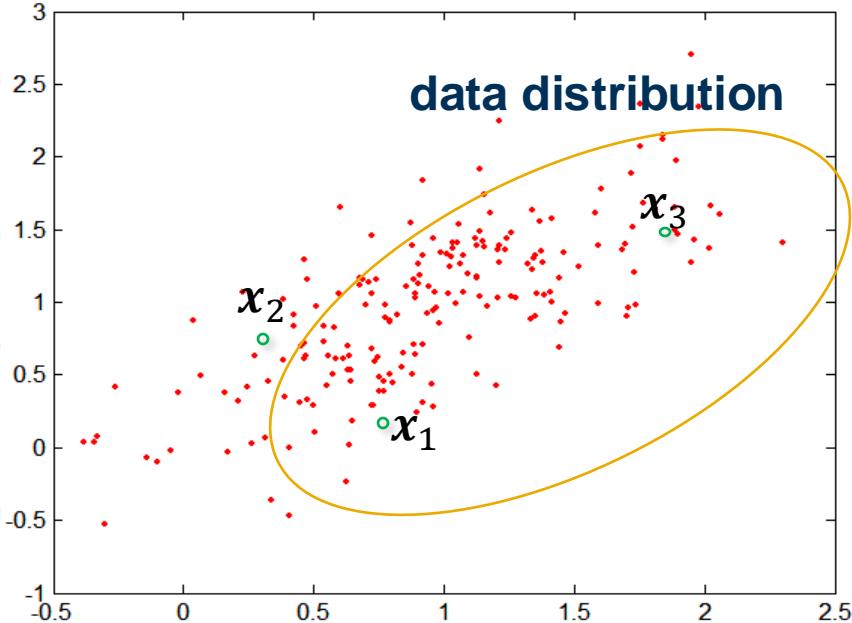
- $C = \begin{bmatrix} \sigma(x, x) & \sigma(x, y) \\ \sigma(y, x) & \sigma(y, y) \end{bmatrix}$

- In general, we can calculate the covariance matrix as follows: $C = \frac{1}{N-1} \sum_{i=1}^N (X_i - \bar{X})(X_i - \bar{X})^T$

- If the mean is zero, we can use the semi-definite matrix to calculate $C = \frac{XX^T}{N-1}$

Proximity Measures

Mahalanobis Distance



$$x_1 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, x_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, x_3 = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}$$

$$\text{Suppose that } \Sigma \text{ is given } \Sigma = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}, \Sigma^{-1} = \begin{bmatrix} 6 & -4 \\ -4 & 6 \end{bmatrix}$$

Note that Σ is computed over the entire dataset, not just the three points

$$d(x_1, x_2) = (x_1 - x_2)^T \Sigma^{-1} (x_1 - x_2)$$

$$= [0.5 \quad -0.5] \begin{bmatrix} 6 & -4 \\ -4 & 6 \end{bmatrix} \begin{bmatrix} +0.5 \\ -0.5 \end{bmatrix} = 5$$

$$d(x_1, x_3) = (x_1 - x_3)^T \Sigma^{-1} (x_1 - x_3)$$

$$= [-1.0 \quad -1.0] \begin{bmatrix} 6 & -4 \\ -4 & 6 \end{bmatrix} \begin{bmatrix} -1.0 \\ -1.0 \end{bmatrix} = 4$$

- Note that although x_2 has a closer direct Euclidean distance to x_1 than x_3 , $d(x_1, x_2)$ is a larger Mahalanobis distance since x_2 is not inline with the overall data distribution.

Appendix A: Notations

- x_i : a single feature
- \mathbf{x}_i : feature vector (a data sample)
- $\mathbf{x}_{:,i}$: feature vector of all data samples
- \mathbf{X} : matrix of feature vectors (dataset)
- N : number of data samples
- m : degree of polynomial
- P : number of features in a feature vector
- θ_i : a single model coefficient (parameter)
- $\boldsymbol{\theta}$: coefficient vector
- ε : error margin
- α : learning rate
- γ : bias factor
- Bold letter/symbol: vector
- Bold capital letters/symbol: matrix