

Lecture 1: Introduction and Linear Algebra Overview

Pang-Ning Tan

Department of Computer Science & Engineering
Michigan State University

August 28, 2019

Outline

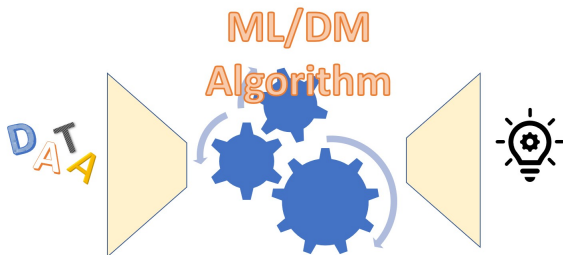
1 Learning from Data

2 Linear Algebra

- Vectors and Their Operations

Learning from Data

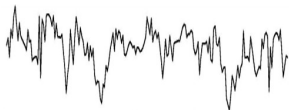
- From Wikipedia:
 - Machine Learning: *The scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead.*
 - Data Science: *A multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data.*
 - Data Mining: *The process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems.*



Learning from Data

Input data can be of varying types

Sensor time series



Surveillance video streams



GPS trajectories from mobile devices



Smart card



Social media



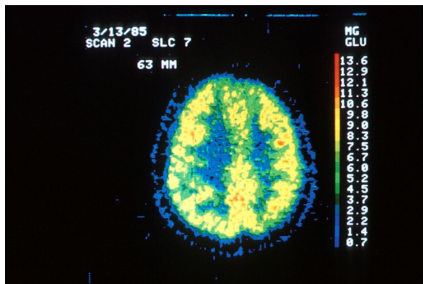
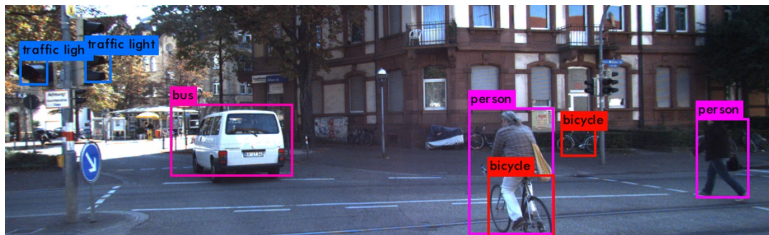
Structured data



But, often times, the same family of algorithms can be applied (with minor modification)

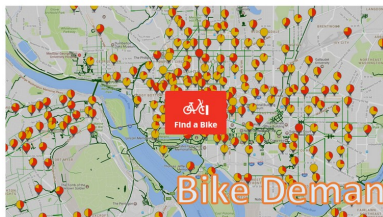
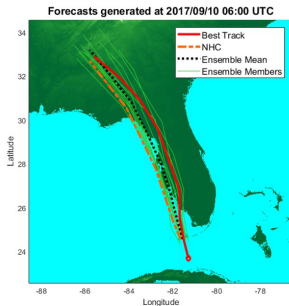
Examples of Learning Tasks

Classification



Examples of Learning Tasks

Regression



Bike Demand Forecasting

Examples of Learning Tasks

Recommendation/Ranking

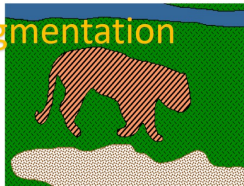
| |  |  |  |  |  |
|---|---|---|---|---|--|
|  | | ** | ? | **** | ? |
|  | | *** | ** | ? | ***** |
|  | | * | ** | ***** | ? |
|  | | ? | ? | *** | *** |

Examples of Learning Tasks

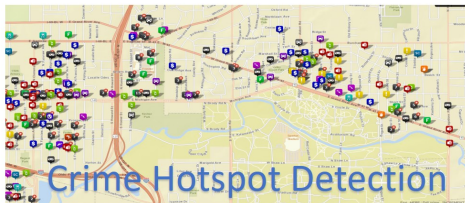
Cluster analysis



Image Segmentation



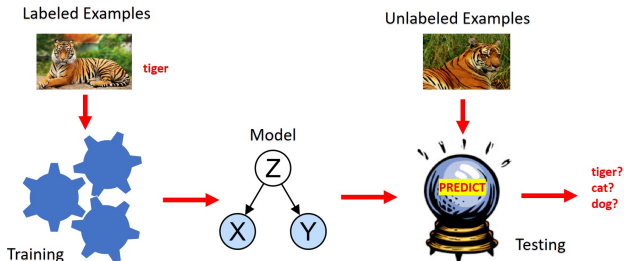
<https://ai.stanford.edu/~syyeung/cvweb/Pictures3/segmentation.png>



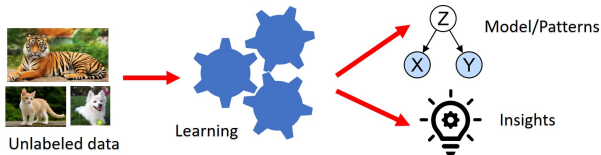
<https://www.lansingmi.gov/569/Crime-Mapping>

Learning Paradigms

- Supervised: Use of labeled examples to guide learning process.

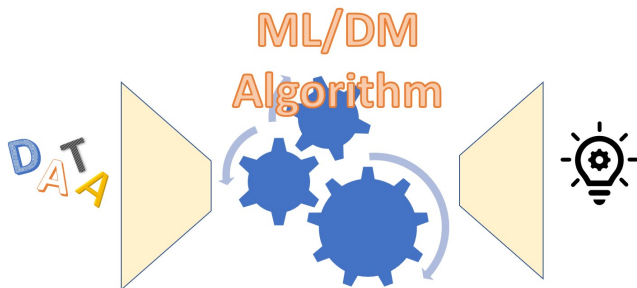


- Unsupervised: No labeled examples to guide learning process.



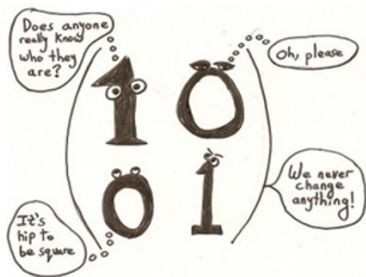
- Others: reinforcement learning, semi-supervised learning, etc.

What is this Class About?



- Goal is to provide the mathematical foundations needed to understand the inner workings of various learning algorithms
 - How and why does an algorithm work?
 - Why does it fail?
- Course is NOT ABOUT how to use tools to do data analysis
 - If this is your interest, you should enroll in 4XX courses (e.g., CSE 402, 404, 482)

Linear Algebra



Why Linear Algebra?

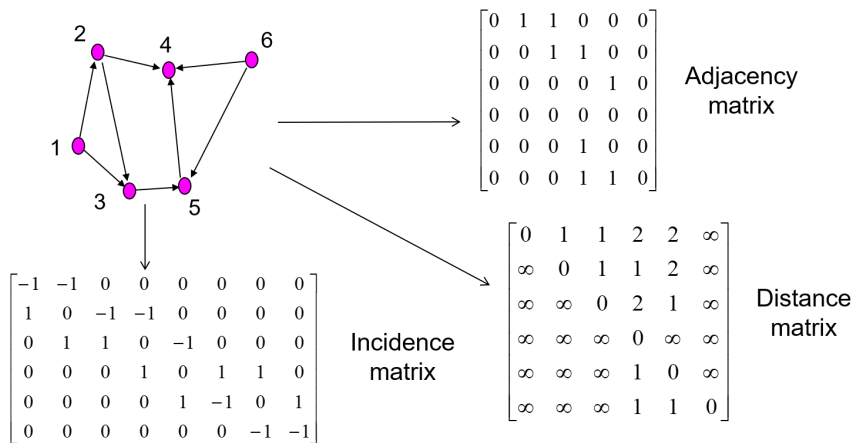
Many data sets can be represented as a data matrix, where the rows are the data objects and the columns are attributes/features of the objects.

| Relation: pima_diabetes | | | | | | | | | |
|-------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|----------------|------------------|
| No. | preg Numeric | plas Numeric | pres Numeric | skin Numeric | insu Numeric | mass Numeric | pedi Numeric | age Numeric | class Nominal |
| 1 | 6.0 | 148.0 | 72.0 | 35.0 | 0.0 | 33.6 | 0.627 | 50.0 | tested_positive |
| 2 | 1.0 | 85.0 | 66.0 | 29.0 | 0.0 | 26.6 | 0.351 | 31.0 | tested_negative |
| 3 | 8.0 | 183.0 | 64.0 | 0.0 | 0.0 | 23.3 | 0.672 | 32.0 | tested_positive |
| 4 | 1.0 | 89.0 | 66.0 | 23.0 | 94.0 | 28.1 | 0.167 | 21.0 | tested_negative |
| 5 | 0.0 | 137.0 | 40.0 | 35.0 | 168.0 | 43.1 | 2.288 | 33.0 | tested_positive |
| 6 | 5.0 | 116.0 | 74.0 | 0.0 | 0.0 | 25.6 | 0.201 | 30.0 | tested_negative |
| 7 | 3.0 | 78.0 | 50.0 | 32.0 | 88.0 | 31.0 | 0.248 | 26.0 | tested_positive |
| 8 | 10.0 | 115.0 | 0.0 | 0.0 | 0.0 | 35.3 | 0.134 | 29.0 | tested_negative |
| 9 | 2.0 | 197.0 | 70.0 | 45.0 | 543.0 | 30.5 | 0.158 | 53.0 | tested_positive |

Figure: A sample of the Pima Indian diabetes dataset (available from <https://www.kaggle.com/uciml/pima-indians-diabetes-database>).

Why Linear Algebra?

Graph data can also be represented as matrices.



Why Linear Algebra?

Given a data matrix \mathbf{X} , we can also compute its covariance matrix and kernel (similarity) matrix

\mathbf{X}

| | | |
|--------|--------|--------|
| 0.8147 | 0.0975 | 0.1576 |
| 0.9058 | 0.2785 | 0.9706 |
| 0.1270 | 0.5469 | 0.9572 |
| 0.9134 | 0.9575 | 0.4854 |
| 0.6324 | 0.9649 | 0.8003 |

Covariance matrix

| | | |
|----------|----------|----------|
| 0.10792 | -0.01127 | -0.05231 |
| -0.01127 | 0.15370 | 0.03194 |
| -0.05231 | 0.03194 | 0.12158 |

$$C(i, j) = \frac{1}{n-1} \sum_{k=1}^n (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)$$

Kernel (similarity) matrix

| | | | | |
|--------|--------|--------|--------|--------|
| 1.0000 | 0.6624 | 0.4625 | 0.6049 | 0.4950 |
| 0.6624 | 1.0000 | 0.6715 | 0.6645 | 0.7137 |
| 0.4625 | 0.6715 | 1.0000 | 0.5530 | 0.7658 |
| 0.6049 | 0.6645 | 0.5530 | 1.0000 | 0.9007 |
| 0.4950 | 0.7137 | 0.7658 | 0.9007 | 1.0000 |

$$K(i, j) = \exp \left[- \frac{\sum_{k=1}^d (x_{ik} - x_{jk})^2}{2\sigma^2} \right]$$

Many machine learning algorithms take as input either the raw data matrix, covariance matrix, or kernel/similarity matrix

Why Linear Algebra?

Many data mining and machine learning tasks can be cast into matrix computation problems.

Movie recommendation as matrix completion problem.

| | Movie | | | | |
|------|-------|---|---|---|---|
| User | 1 | 5 | ? | 4 | 2 |
| | ? | 2 | 3 | ? | 4 |
| | 1 | 1 | ? | 5 | ? |
| | 3 | 5 | 4 | 2 | ? |
| | ? | ? | 5 | ? | 5 |

?: Unrated movie

Matrix completion aims to impute the missing values in a matrix (by assuming the matrix has certain properties, e.g., low rank)

Why Linear Algebra?

Many data mining and machine learning tasks can be cast into matrix computation problems.

Document/word clustering as a matrix factorization problem.

$$\begin{array}{c} \text{Word} \end{array} \begin{array}{c} \text{Document} \\ \begin{bmatrix} 0 & 3 & 0 & 2 & 0 \\ 0 & 6 & 3 & 1 & 4 \\ 1 & 0 & 0 & 5 & 0 \\ 3 & 5 & 2 & 2 & 0 \\ 0 & 0 & 3 & 6 & 3 \\ 2 & 3 & 0 & 0 & 2 \end{bmatrix} \end{array} \longrightarrow \begin{array}{c} \begin{bmatrix} 0.88 & 0.10 \\ 2.06 & 1.03 \\ 0.82 & -1.41 \\ 1.64 & 0.63 \\ 1.55 & -1.79 \\ 0.91 & 0.77 \end{bmatrix} \\ \text{Word} \times \text{Topic} \end{array} \times \begin{array}{c} \begin{bmatrix} 0.65 & 0.27 \\ 2.23 & 1.64 \\ 1.22 & -0.14 \\ 1.77 & -2.11 \\ 1.27 & 0.04 \end{bmatrix}^T \\ \text{Document} \times \text{Topic} \end{array}$$

Let $\mathbf{x} \in \mathbb{R}^d$ be a d -dimensional column vector:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_d \end{bmatrix}$$

where each x_i is an element of the vector

By convention, a vector is represented

- Using a bold lower-case letter.
- As a column vector (unless noted otherwise); can also be denoted (in Matlab notation) as $[x_1; x_2; \dots; x_d]$.

Elements of a Vector

Each element is drawn from some given field¹:

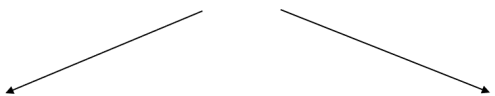
- $\mathbf{x} \in \mathbb{R}^d$: elements of the vector are real-valued.
 - Example: $\mathbf{x} = [1.2; 2.3; 0; -2.5]$
- $\mathbf{x} \in \mathbb{R}_+^d$: elements are non-negative real-values, i.e., $\mathbf{x} \in [0, \infty)^d$.
 - Example: $\mathbf{x} = [1.2; 2.3; 0; 2.5]$
- $\mathbf{x} \in \mathbb{C}^d$: elements of the vector are complex-valued.
 - Example: $\mathbf{x} = [1.2 + i2.5; 2.3 + i0; -2.5 - i1.5]$
- $\mathbf{x} \in [0, 1]^d$: elements are real values between 0 and 1.
 - Example: $\mathbf{x} = [0.2; 1; 0; 0.64]$
- $\mathbf{x} \in \{0, 1\}^d$: elements are binary integers, either 0 or 1.
 - Example: $\mathbf{x} = [0; 1; 1; 0]$

¹A field is a set of values with well-defined addition and multiplication operations.

Vector Representation of Data

Converting data frame (e.g., Excel spreadsheet) into vectors.

| | Age | Height | Weight |
|------|-----|--------|--------|
| John | 28 | 6.1 | 176 |
| Mary | 35 | 5.7 | 130 |


$$\mathbf{john} = \begin{bmatrix} 28 \\ 6.1 \\ 176 \end{bmatrix}$$

$$\mathbf{age} = \begin{bmatrix} 28 \\ 35 \end{bmatrix} \quad \mathbf{height} = \begin{bmatrix} 6.1 \\ 5.7 \end{bmatrix}$$

$$\mathbf{mary} = \begin{bmatrix} 35 \\ 5.7 \\ 130 \end{bmatrix}$$

$$\mathbf{weight} = \begin{bmatrix} 176 \\ 130 \end{bmatrix}$$

Vector Representation of Data

Converting data frame (e.g., Excel spreadsheet) into vectors.

| | Age | Height | Weight | Gender |
|------|-----|--------|--------|--------|
| John | 28 | 6.1 | 176 | Male |
| Mary | 35 | 5.7 | 130 | Female |

Create a new
element for each
discrete value

$$\mathbf{john} = \begin{bmatrix} 28 \\ 6.1 \\ 176 \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix}$$

$$\mathbf{mary} = \begin{bmatrix} 35 \\ 5.7 \\ 130 \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix}$$

$$\mathbf{male} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\mathbf{female} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

This transformation is known as one-hot encoding.

“Special” Vectors

- $\mathbf{x} = \mathbf{0}$: a null vector, whose elements are all equal to zero
- $\mathbf{x} = \mathbf{1}$: a vector whose elements are all equal to one
- $\mathbf{x} = \mathbf{e}_i$: a canonical vector whose i -th element is equal to 1, but 0 elsewhere

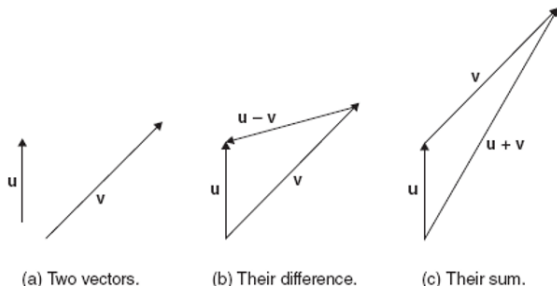
$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ \dots \\ 0 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ \dots \\ 0 \end{bmatrix}, \quad \mathbf{e}_n = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 1 \end{bmatrix}$$

Vector Addition and Subtraction

- Let $\mathbf{u} = \begin{bmatrix} 1.7 \\ 19 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} 2.3 \\ -8 \end{bmatrix}$
- Addition/Subtraction:

$$\mathbf{u} + \mathbf{v} = \begin{bmatrix} 4.0 \\ 11 \end{bmatrix}, \quad \mathbf{v} - \mathbf{u} = \begin{bmatrix} 0.6 \\ -27 \end{bmatrix}$$

- Geometric interpretation:



Vector Transpose

- If $\mathbf{u} = \begin{bmatrix} 1.7 \\ 19 \end{bmatrix}$, then its transpose, $\mathbf{u}^T = [1.7 \ 19]$, is a row vector.

| | Age | Height | Weight |
|------|-----|--------|--------|
| John | 28 | 6.1 | 176 |
| Mary | 35 | 5.7 | 130 |

$$\mathbf{john} = \begin{bmatrix} 28 \\ 6.1 \\ 176 \end{bmatrix}$$

$$\mathbf{mary} = \begin{bmatrix} 35 \\ 5.7 \\ 130 \end{bmatrix}$$

What is $(\mathbf{u}^T)^T$?

$$\mathbf{john}^T = [28 \ 6.1 \ 176]$$

$$\mathbf{mary}^T = [35 \ 5.7 \ 130]$$

Vector Norms

- Let $\mathbf{u} = \begin{bmatrix} 1.7 \\ 0 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} 2.3 \\ -8 \end{bmatrix}$
- Vector norm (“length”): $\|\mathbf{u}\|_p = \sqrt[p]{\sum_i |u_i|^p}$ (also known as ℓ_p -norm)
 - ℓ_2 -norm (Euclidean norm)

$$\|\mathbf{u}\|_2 = \sqrt{|1.7|^2} = 1.7, \quad \|\mathbf{v}\|_2 = \sqrt{|2.3|^2 + |-8|^2} = 8.32$$

- ℓ_1 norm (sum of absolute values)

$$\|\mathbf{u}\|_1 = 1.7, \quad \|\mathbf{v}\|_1 = |2.3| + |-8| = 10.3$$

- ℓ_∞ norm (maximum absolute value)

$$\|\mathbf{u}\|_\infty = \max(|1.7|, 0) = 1.7, \quad \|\mathbf{v}\|_\infty = \max(|2.3|, |-8|) = 8$$

- ℓ_0 “norm” (# non-zero elements); not a proper norm

$$\|\mathbf{u}\|_0 = 1, \quad \|\mathbf{v}\|_0 = 2$$

- Proof:

$$\|\mathbf{x}\|_p = \left[\sum_i |x_i|^p \right]^{\frac{1}{p}} \leq \left[\sum_{i=1}^d \left(\max_j |x_j| \right)^p \right]^{\frac{1}{p}} = \left[d \left(\max_j |x_j| \right)^p \right]^{\frac{1}{p}}$$

Therefore

$$\|\mathbf{x}\|_p \leq d^{\frac{1}{p}} \max_j |x_j|$$

Furthermore

$$\max_j |x_j| \leq \|\mathbf{x}\|_p \quad (\text{Proof left as exercise})$$

Therefore

$$\max_j |x_j| \leq \|\mathbf{x}\|_p \leq d^{\frac{1}{p}} \max_j |x_j|$$

Proof follows from taking the limit $p \rightarrow \infty$ ($d^{\frac{1}{p}} \rightarrow 1$)

Properties of Vector Norm

- A function $\|\cdot\| : \mathbb{R}^d \rightarrow \mathbb{R}$ is called a vector norm if it satisfies the following properties:
 - 1 $\|\mathbf{x}\| \geq 0$, where $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = \mathbf{0}$
 - 2 $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|$ for any scalar $\alpha \in \mathbb{R}$
 - 3 $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ (triangle inequality)
- Exercise: Which property is not satisfied by ℓ_0 “norm”?

Vector Norms in Machine Learning

- As a distance or similarity measure between data points

| | Age | Height | Weight |
|------|-----|--------|--------|
| John | 28 | 6.1 | 176 |
| Mary | 35 | 5.7 | 130 |

- ℓ_2 distance

$$\ell_2(\text{John}, \text{Mary}) = \sqrt{7^2 + 0.4^2 + 46^2} = 46.53$$

- Gaussian radial basis function (rbf) similarity

$$S(\mathbf{x}, \mathbf{y}) = \exp[-\gamma \|\mathbf{x} - \mathbf{y}\|_2^2]$$

$$S(\text{John}, \text{Mary}) = \exp[-\gamma \times 2165.16]$$

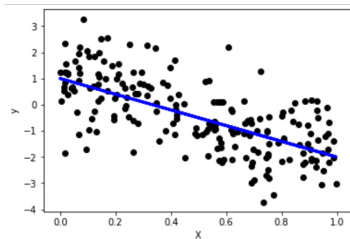
(γ is known as the kernel parameter)

Vector Norms in Machine Learning

- As a measure of model sparsity
- For example, suppose we want to fit a linear regression model,

$$y = \mathbf{w}^T \mathbf{x} + w_0 = w_5 x_5 + w_4 x_4 + w_3 x_3 + w_2 x_2 + w_1 x + w_0,$$

to the dataset shown below.



| X | X2 | X3 | X4 | X5 | y |
|----------|-----------|-----------|-----------|-----------|-----------|
| 0.417022 | 0.273485 | 0.132734 | 0.053301 | 0.015921 | -0.173245 |
| 0.720324 | 0.335692 | 0.176086 | 0.088807 | 0.049355 | -0.542593 |
| 0.000114 | -0.021070 | -0.016158 | -0.004407 | -0.011724 | 1.232151 |
| 0.302333 | 0.108248 | 0.073673 | 0.049165 | 0.019401 | 0.775554 |
| 0.146756 | 0.107994 | 0.040678 | 0.016110 | -0.006559 | 0.249616 |
| 0.092339 | -0.045892 | -0.040553 | -0.019412 | -0.014869 | -1.711854 |
| 0.186260 | 0.162923 | 0.064954 | 0.011052 | 0.009037 | 1.480044 |
| 0.345561 | 0.142332 | 0.062260 | 0.022829 | 0.010727 | 2.150297 |

- Ground truth: $y = 3x + 1$
(x_2, x_3, x_4 , and x_5 are irrelevant features, i.e., unrelated to y)

Vector Norms in Machine Learning

- Regression results (with different model complexities)

| Model | Train error | Test error | $\ \mathbf{w}\ $ |
|--|-------------|------------|------------------|
| -3.24 X + 1.08 | 0.891873 | 1.047626 | 4.322954 |
| -5.90 X + 5.92 X ² + 1.00 | 0.856157 | 1.087601 | 12.817040 |
| -6.22 X + -2.30 X ² + 17.14 X ³ + 1.08 | 0.834238 | 1.094661 | 26.744867 |
| -7.16 X + 0.93 X ² + 8.39 X ³ + 11.85 X ⁴ + 1.12 | 0.825722 | 1.128861 | 29.453660 |
| -7.16 X + 4.50 X ² + 3.52 X ³ + -6.55 X ⁴ + 25.68 X ⁵ + 1.20 | 0.799399 | 1.146546 | 48.614927 |

- Train error: prediction error on the dataset used to fit the model
- Test error: prediction error on a separately withheld dataset (which is not used to fit the model)
- Overfitting: model has small training error but large test error
- $\|\mathbf{w}\|$ refers to the ℓ_1 norm of the regression coefficients and can be used to measure model sparsity
 - A sparse model has smaller $\|\mathbf{w}\|$ and will less likely to overfit the training data

Unit Vectors

- A vector whose norm is equal to one.
- Example: Unit vectors for a 3-d Cartesian coordinate system:

$$\hat{\mathbf{i}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \hat{\mathbf{j}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \hat{\mathbf{k}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- Given any vector \mathbf{u} , the unit vector in the direction of \mathbf{u} is

$$\hat{\mathbf{u}} = \frac{\mathbf{u}}{\|\mathbf{u}\|}$$

Dot Product of Vectors

- Algebraic definition: $\mathbf{u} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{v} = \sum_i u_i v_i$

- Example: let $\mathbf{u} = \begin{bmatrix} 1.7 \\ 19 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} 2.3 \\ -8 \end{bmatrix}$

$$\begin{aligned} \mathbf{u} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{v} &= \begin{bmatrix} 1.7 \\ 19 \end{bmatrix}^T \begin{bmatrix} 2.3 \\ -8 \end{bmatrix} \\ &= \begin{bmatrix} 1.7 & 19 \end{bmatrix} \begin{bmatrix} 2.3 \\ -8 \end{bmatrix} \\ &= 1.7 \times 2.3 + 19 \times (-8) \\ &= -148.9 \end{aligned}$$

Dot Product of Vectors

Some useful properties:

- Positivity: $\mathbf{u} \cdot \mathbf{u} \geq 0$ (equality holds only when $\mathbf{u} = \mathbf{0}$).
- Symmetry: $\mathbf{u} \cdot \mathbf{v} = \mathbf{v} \cdot \mathbf{u}$.
- Scalar multiplication:

$$(\alpha \mathbf{u}) \cdot \mathbf{v} = \mathbf{u} \cdot (\alpha \mathbf{v}) = \alpha(\mathbf{u} \cdot \mathbf{v}),$$

where α is a scalar

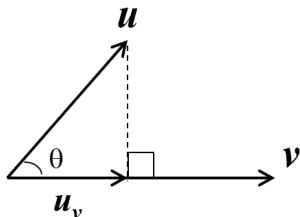
- Distributive: $\mathbf{u} \cdot (\mathbf{v} + \mathbf{w}) = \mathbf{u} \cdot \mathbf{v} + \mathbf{u} \cdot \mathbf{w}$

Dot Product of Vectors

- Geometric Interpretation in Euclidean space:

$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \|\mathbf{v}\| \cos \theta = \|\mathbf{v}\| \|\mathbf{u}_v\|,$$

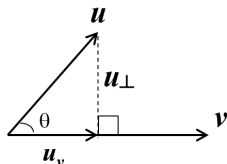
where θ is the angle between \mathbf{u} and \mathbf{v} , $\|\cdot\|$ is the Euclidean norm, and \mathbf{u}_v is the projection of \mathbf{u} onto \mathbf{v}



- If \mathbf{u} and \mathbf{v} are orthogonal ($\theta = 90^\circ$), then $\mathbf{u} \cdot \mathbf{v} = 0$.

Orthogonal Projection

- Given a vector \mathbf{u}



- \mathbf{u}_v is the component of \mathbf{u} in the direction of \mathbf{v}

$$\mathbf{u}_v = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{v}\|} \hat{\mathbf{v}} = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{v}\|} \frac{\mathbf{v}}{\|\mathbf{v}\|} = \left(\frac{\mathbf{u} \cdot \mathbf{v}}{\mathbf{v} \cdot \mathbf{v}} \right) \mathbf{v}$$

where $\hat{\mathbf{v}}$ is the unit vector along the direction of \mathbf{v} .

- \mathbf{u}_\perp is the component of \mathbf{u} orthogonal to \mathbf{v}

$$\mathbf{u}_\perp = \mathbf{u} - \mathbf{u}_v = \mathbf{u} - \left(\frac{\mathbf{u} \cdot \mathbf{v}}{\mathbf{v} \cdot \mathbf{v}} \right) \mathbf{v}$$

Cauchy Schwarz Inequality

$$|\mathbf{x}^T \mathbf{y}| \leq \|\mathbf{x}\| \|\mathbf{y}\|$$

- Simple proof:

- By definition: $\mathbf{x}^T \mathbf{y} = \mathbf{x} \cdot \mathbf{y} = \|\mathbf{x}\| \|\mathbf{y}\| \cos \theta$
- Since $|\cos \theta| \leq 1$, therefore $|\mathbf{x}^T \mathbf{y}| \leq \|\mathbf{x}\| \|\mathbf{y}\|$

- Another proof:

- Since $\mathbf{u} = \mathbf{u}_v + \mathbf{u}_\perp$ and \mathbf{u}_v is orthogonal to \mathbf{u}_\perp

$$\begin{aligned} \|\mathbf{u}\|^2 &= \|\mathbf{u}_v\|^2 + \|\mathbf{u}_\perp\|^2 = \left| \frac{\mathbf{u}^T \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \right|^2 \|\mathbf{v}\|^2 + \|\mathbf{u}_\perp\|^2 \\ &\geq \left| \frac{\mathbf{u}^T \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \right|^2 \|\mathbf{v}\|^2 = \frac{|\mathbf{u}^T \mathbf{v}|^2}{\|\mathbf{v}\|^2} \quad (\text{where } \mathbf{v}^T \mathbf{v} = \|\mathbf{v}\|^2) \\ \therefore \|\mathbf{u}\|^2 \|\mathbf{v}\|^2 &\geq |\mathbf{u}^T \mathbf{v}|^2 \end{aligned}$$

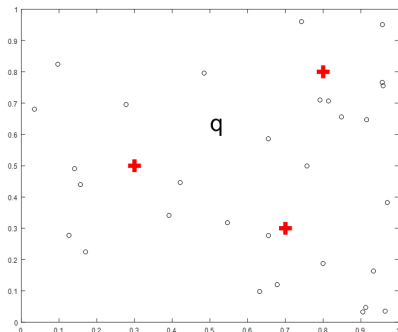
Triangle Inequality

$$\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$$

• Proof:

$$\begin{aligned}\|\mathbf{x} + \mathbf{y}\|^2 &= (\mathbf{x} + \mathbf{y})^T (\mathbf{x} + \mathbf{y}) \\&= \|\mathbf{x}\|^2 + 2\mathbf{x}^T \mathbf{y} + \|\mathbf{y}\|^2 \\&\leq \|\mathbf{x}\|^2 + 2\|\mathbf{x}\|\|\mathbf{y}\| + \|\mathbf{y}\|^2 \quad (\text{Cauchy-Schwarz inequality}) \\&= (\|\mathbf{x}\| + \|\mathbf{y}\|)^2 \\ \therefore \|\mathbf{x} + \mathbf{y}\| &\leq \|\mathbf{x}\| + \|\mathbf{y}\|\end{aligned}$$

Application to Similarity Search



+: Landmark objects

q: query object

Given a collection of 100 million objects in Ω , how do we quickly identify a subset of objects Q that are similar to an input query object (q)?

$$Q = \{o \mid \forall o \in \Omega : d(o, q) < \epsilon\}$$

Let $R = \{r_1, r_2, \dots, r_k\}$ be a subset of the objects (landmark points)

Assume $d(o, r)$ has been precomputed

Triangle inequality can be used to improve search efficiency:

- ① $d(q, o) \leq d(q, r) + d(r, o)$
- ② $d(q, o) \geq |d(q, r) - d(r, o)|$

Numpy Programming

- Numpy
 - Stands for numerical Python, a Python library package to support numerical computations.
 - ndarray: the basic data structure in Numpy
 - provides a suite of functions that can efficiently manipulate elements of the ndarray.
- Python coding practice exercises:
Go to <http://www.cse.msu.edu/~ptan/dmbook/software>
and go over modules 0, 1, and 2.