Mathematical Preliminaries

Linear Algebra

UNIVERSITY OF LIVERPOOL

Linear algebra

- In Data Mining, we will represent data points using **vectors** — ordered sets of coordinates (corresponding to various attributes/features)

- The branch of mathematics that concerns with such coordinated representations is called **linear algebra**

- Reference: Chapter 02 of the MML book [https://mml-book.github.io/book/mml-book.pdf]

Vectors

- We will denote a vectors by $\overline{X}, \overline{Y}, \overline{W}, \ldots$ (uppercase letters with a bar)

- We will use column vectors throughout this module (transposed by T when written as row vectors) e.g.

$$\overline{X} = (3.2, -9.1, 0.1)^T$$

- $\overline{X} \in \mathbb{R}^d$ means that $\overline{X}$ is a d-dimensional vector with real coordinates

Matrices

- We obtain matrices by arranging a collection of vectors by columns or rows.
- Similarly to the vectors, we use uppercase letters with a bar to denote matrices such as $\overline{M}$
- $\overline{M} \in \mathbb{R}^{n \times m}$ means that $\overline{M}$ is a matrix with n rows and m columns
- When $n = m$ we say M is **square**
- We denote the (i, j) element of $\overline{M}$ by $\overline{M}_{i,j}$
- If $\overline{M}_{i,j} = \overline{M}_{j,i}$ for all i and j, we say $\overline{M}$ is **symmetric**. Otherwise, $\overline{M}$ is asymmetric

Vector arithmetic

- Given two vectors $\overline{X}, \overline{Y} \in \mathbb{R}^d$, where $\overline{X} = (x_1, \ldots, x_d)^T$ and $\overline{Y} = (y_1, \ldots, y_d)^T$
- Their **addition** is given by the vector $\overline{Z} = (z_1, \ldots, z_d)^T$ where the i-th element z_i is given by $z_i = x_i + y_i$
- Their **inner-product** (also known as **dot product**) is defined as

$$\overline{X}^T\overline{Y} = \sum_{i=1}^{d} x_i y_i$$

- Their **outer-product** $\overline{X}\overline{Y}^T$ is defined as the matrix $\overline{M} \in \mathbb{R}^{d \times d}$, where $\overline{M}_{i,j} = x_i \cdot y_j$

Matrix arithmetic

- Matrices of the same shape (number of rows and columns) can be **added** element-wise

$$\overline{A} + \overline{B} = \overline{C}\text{, where } \overline{C}_{i,j} = \overline{A}_{i,j} + \overline{B}_{i,j}$$

- Matrices can be **multiplied** if the number of columns of the first matrix is equal to the number of rows of the second matrix. Let $\overline{A} \in \mathbb{R}^{n \times m}$ and $\overline{B} = \mathbb{R}^{m \times d}$, then the matrix $\overline{C} = \overline{A}\overline{B}$ has n rows and d columns,

$$\overline{C}_{i,j} = \sum_{k=1}^{m} \overline{A}_{i,k} \overline{B}_{k,j}$$

Transpose and Inverse

- The **transpose** of a matrix $\overline{A} \in \mathbb{R}^{n \times d}$ is denoted by $\overline{A}^T$ and is a matrix from $\mathbb{R}^{d \times n}$, where $\overline{A}^T_{i,k} = \overline{A}_{k,i}$

- $(\overline{A}\overline{B})^T = \overline{B}^T \overline{A}^T$

- The **inverse** of a square matrix $\overline{A} \in \mathbb{R}^{n \times n}$ is denoted by $\overline{A}^{-1}$ and satisfies $\overline{A}\overline{A}^{-1} = \overline{A}^{-1}\overline{A} = I$, where $I \in \mathbb{R}^{n \times n}$ is the **unit matrix** (all diagonal elements are set to 1 and non-diagonal elements are set to 0)

Linear independence

- Let us consider a vector $\overline{V}$ formed as the linearly-weighted sum of a set of vectors $\{\overline{X}_1, \ldots, \overline{X}_k\}$ with respective coefficients $\lambda_1, \ldots, \lambda_k$ as follows:

$$\overline{V} = \lambda_1 \overline{X}_1 + \ldots + \lambda_k \overline{X}_k = \sum_{i=1}^{k} \lambda_i \overline{X}_i$$

- $\overline{V}$ is called a **linear combination** of $\overline{X}_1, \ldots, \overline{X}_k$

- Vectors $\overline{X}_1, \ldots, \overline{X}_k$ are called **linearly dependent** if there exists $\lambda_1, \ldots, \lambda_k$, not all zero, such that

$$\overline{0} = \lambda_1 \overline{X}_1 + \ldots + \lambda_k \overline{X}_k$$

- Otherwise $\overline{X}_1, \ldots, \overline{X}_k$ are called **linearly independent**

Rank

- The number of linear independent columns of a matrix $\bar{A} \in \mathbb{R}^{m \times n}$ $(m \leq n)$ equals the number of linearly independent rows and is called the **rank** of $\bar{A}$ is denoted by $\text{rank}(\bar{A})$

- $\text{rank}(\bar{A}) \leq \min\{m, n\} = m$

- If $\text{rank}(\bar{A}) = m$, then $\bar{A}$ is said to be **full-rank**, otherwise **rank-deficient**.

- Only full-rank square matrices are invertible.

Matrix trace

The sum of diagonal elements is called the trace of the matrix. Specifically,

$$\mathrm{tr}(\bar{A}) = \sum_i \bar{A}_{i,i}$$

Example

$$\bar{A} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \mathrm{tr}(\bar{A}) = 2$$

Eigenvalues and eigenvectors

Let $\overline{A} \in \mathbb{R}^{n \times n}$ be a square matrix.

A non-zero vector $\overline{X} \in \mathbb{R}^n$ is an **eigenvector** of $\overline{A}$ if

$$\overline{A}\overline{X} = \lambda \overline{X}$$

for some $\lambda \in \mathbb{R}$, which is called **eigenvalue** of $\overline{A}$ corresponding to $\overline{X}$.

Mathematical Preliminaries

Differential Calculas

Derivatives of basic functions

$\frac{d}{dx}a = 0$, where a is a constant (i.e., does not depend on x)

$\frac{d}{dx}x^a = a \cdot x^{a-1}$

$\frac{d}{dx}e^x = e^x$, where $e \approx 2.71$ is Euler's number

$\frac{d}{dx}\log(x) = \frac{1}{x}$, where $x > 0$

$\frac{d}{dx}\sin(x) = \cos(x)$

$\frac{d}{dx}\cos(x) = -\sin(x)$

Note: $\frac{d}{dx}f(x)$ is the same as $f'(x)$

Differentiation rules

Sum rule: $(\alpha f + \beta g)' = \alpha f' + \beta g'$

Product rule: $(fg)' = f'g + fg'$

Quotient rule: $\left(\dfrac{f}{g}\right)' = \dfrac{f'g - fg'}{g^2}$

Chain rule: If $f(x) = h(g(x))$, then $f'(x) = h'(g(x)) \cdot g'(x) = \dfrac{d}{dg(x)}h \cdot \dfrac{d}{dx}g$

Partial derivative

A **partial derivative** of a function of several variables is its derivative with respect to one of those variables, with the others held constant.

Example

$$f(x, y) = 5x + y^2$$

$$\frac{\partial f}{\partial x} = 5 \qquad \frac{\partial f}{\partial y} = 2y \qquad \nabla_{(x,y)} f = \left(5, \; 2y\right)^T$$

Reference

Chapter 6 of the MML book
[https://mml-book.github.io/book/mml-book.pdf]

Mathematical Preliminaries

Optimisation

Continuous optimisation

- Unconstrained optimisation
- Constrained optimisation

- Reference: Chapter 7 of the MML book [https://mml-book.github.io/book/mml-book.pdf]

Unconstrained optimisation: Gradient Descent method

Problem formulation

$$\text{find } \min_{\overline{X}} f(\overline{X})$$

where

1) $\overline{X} = (x_1, x_2, \ldots, x_d)$ and

2) $f : \mathbb{R}^d \to \mathbb{R}$

3) f is differentiable and we are unable to analytically find a solution in closed form

Unconstrained optimisation: Gradient Descent method

Let $f(\overline{X}) = f(x_1, \ldots, x_d)$ be a function depending on d variables.

The **gradient** of $f(\overline{X})$ is the vector consisting of the partial derivatives of f

$$\nabla_{\overline{X}} f = \frac{\partial f}{\partial \overline{X}} = \left(\frac{\partial f(\overline{X})}{\partial x_1} \quad \frac{\partial f(\overline{X})}{\partial x_2} \quad \cdots \quad \frac{\partial f(\overline{X})}{\partial x_d} \right)^T$$

Unconstrained optimisation: Gradient Descent method

The gradient $\nabla_{\overline{X}} f$ evaluated in a point $\overline{X}_0$ gives a vector that points in the direction of the steepest ascent.

Starting from point $\overline{X}_0$ the function f **decreases** faster if one moves from X_0 in the direction of the **negative gradient** of f at $\overline{X}_0$, i.e. in the direction of the vector $-(\nabla_{\overline{X}} f)(\overline{X}_0)$

This means that for a small step-size $\gamma \geq 0$ the value of the function in point

$$\overline{X}_1 = \overline{X}_0 - \gamma \cdot (\nabla_{\overline{X}} f)(\overline{X}_0)$$

is smaller than in the initial point $\overline{X}_0$, i.e.,

$$f(\overline{X}_1) \leq f(\overline{X}_0)$$

Unconstrained optimisation: Gradient Descent method

Algorithm for finding local minimum of $f(\overline{X}) = f(x_1, \ldots, x_d)$

1. Pick an initial point $\overline{X}_0$

2. Iterate according to

$$\overline{X}_{i+1} = \overline{X}_i - \gamma_i \cdot \big((\nabla_{\overline{X}} f)(\overline{X}_i) \big)$$

For a suitable step-sizes $\gamma_1, \gamma_2, \ldots$, the sequence $f(\overline{X}_0) \geq f(\overline{X}_1) \geq \ldots$ converges to a local minimum.

Moral: gradient of a function is a useful tool for finding local optimal points of a function and is widely used in data mining and machine learning.

Constrained optimisation: method of Lagrange multipliers

Problem formulation

find $\min_{\overline{X}} f(\overline{X})$

Subject to $g(\overline{X}) = 0$

where

1) $\overline{X} = (x_1, x_2, \ldots, x_d)$ and

2) $f : \mathbb{R}^d \to \mathbb{R}$

3) $g : \mathbb{R}^d \to \mathbb{R}$

4) f is differentiable and we are unable to analytically find a solution in closed form

Constrained optimisation: method of Lagrange multipliers

Problem formulation

$$\text{find } \min_{\overline{X}} f(\overline{X})$$

$$\text{Subject to } g(\overline{X}) = 0$$

In order to solve this problem:

1. Form the Lagrangian function $\mathscr{L}(\overline{X}, \lambda) = f(\overline{X}) - \lambda \cdot g(\overline{X})$
2. Find all stationary points $(\overline{X}_0, \lambda_0)$ of $\mathscr{L}(\overline{X}, \lambda)$, i.e. those points for which all partial derivatives of $\mathscr{L}(\overline{X}, \lambda)$ are equal to 0, or equivalently $\nabla_{(\overline{X},\lambda)}\mathscr{L} = \overline{0}$
3. Examine stationary points to find among them a solution to the problem

Mathematical Preliminaries

Probability

Common discrete probability distributions

- **Bernoulli distribution**: models binary outcomes (coin flip).

$P(X = \text{head}) = p$ and $P(X = \text{tail}) = 1 - p$

- **Generalised Bernoulli distribution**: models $k > 2$ outcomes (rolls of a k -sided die)

$P(X = 1) = p_1, P(X = 2) = p_2, \ldots, P(X = k) = p_k$ such that $\sum_{i=1}^{k} p_i = 1$

Common discrete probability distributions

- **Binomial distribution**: models a sequence of multiple flips of a coin

$$P(\text{in } n \text{ flips there are exactly } k \text{ heads}) = \binom{n}{k} p^k (1-p)^{n-k}$$

- **Multinomial distribution**: models a sequence of multiple rolls of a k-sided die for $k > 2$

If there are n rolls and n_i is the number of times the die came up on side i, then the probability of this event is

$$\frac{n!}{\prod_{i=1}^{k} n_i!} \cdot \prod_{i=1}^{k} p_i^{n_i}$$