Timothy Chung

COMP70015
Mathematics for Machine Learning

AUTUMN 2024

Contents

Introduction

My notes try to combine the detail of the professor's informal lecture notes with the breadth of the slides, where the latter covers all the assessed mathematical content. However, I will have added extra material from the listed reference materials in the informal notes.

Overview

Week 1: Discuss ML, brush up skills

Week 2: Critical optimisation concepts: automatic differentiation, convergence, complexity

Week 3: Probabilistic Perspective to ML

Week 4: Probabilistic Perspective to ML (contd.)

Week 5: Focus on Bayes Theorem + Practicals

Week 6: Consolidating learning (PCA study)

Week 7: Two Advanced topics: NN training with adversarial examples, + student selected topic

Part I
Fundamental Concepts in Mathematics for Machine Learning

1
Formalising ML Problem Settings

Note 1.0.1 On The Rigour of A Random Variable

For the beginning of this module, a random variable X is considered to be a function from a sample space Ω to the unit interval $[0, 1]$. However, in Lectures 6 and 7, we use the full definition of a random variable, which is a measurable function from a sample probability space (Ω) to a measurable space (E, ε), involving the concept of σ-algebras and measure spaces.

When the sample space is a continuous space, the function p is a **probability density function** (PDF). When the sample space is a discrete space, the function p is a **probability mass function** (PMF).

We will introduce measure theory later.

1.1 Brief Probability Theory Review

x is a sample from a random variable X.
$P(X = x)$ refers to probability that the random variable X takes on the value x.

1.1.1 Probability Density Function (PDF)

The Probability Density Function (PDF) of a continuous random variable X is a function $f_X(x)$ that describes the relative likelihood for this random variable to take on a given value. The PDF has the following properties:

- $f_X(x) \geq 0$ for all x.
- $\int_{-\infty}^{\infty} f_X(x)\, dx = 1$.

Mathematically, the PDF is defined such that the probability that X lies within a particular interval $[a, b]$ is given by:

$$P(a \leq X \leq b) = \int_a^b f_X(x)\, dx.$$

Example PDF: Normal Distribution

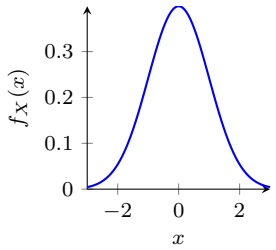


Figure 1.1: Probability Density Function of a standard normal distribution

1.1.2 Cumulative Distribution Function (CDF)

The Cumulative Distribution Function (CDF) of a continuous random variable X is a function $F_X(x)$ that describes the probability that X will take a value less than or equal to x. It is defined as:

$$F_X(x) = P(X \leq x) = \int_{-\infty}^{x} f_X(t)\, dt.$$

The CDF has the following properties:

- $0 \leq F_X(x) \leq 1$ for all x.
- $F_X(x)$ is a non-decreasing function.
- $\lim_{x \to -\infty} F_X(x) = 0$.
- $\lim_{x \to \infty} F_X(x) = 1$.

Example CDF: Sigmoid Approximation

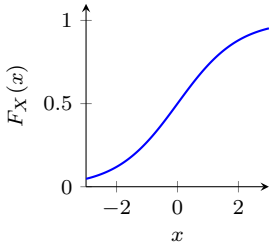


Figure 1.2: Cumulative Distribution Function

1.1.3 Dealing with Data (Single-Feature)

In Machine Learning (ML), we often deal with large datasets from which we aim to extract meaningful insights. It is useful to model these data points as random variables:

$$\{x_1, x_2, \ldots, x_N\}$$

We typically model these samples as random variables:

$$\{x_1 \sim X_1, x_2 \sim X_2, \ldots, x_N \sim X_N\}$$

1.1.4 Vectors of Random Variables

When dealing with joint random variables, we often represent them as vectors:

$$\mathbf{X} = (X_1, X_2, X_3) \quad \text{with joint distribution} \quad p_{X_1,X_2,X_3}(x_1, x_2, x_3)$$

For a vector of random variables $\mathbf{X}$ in $\mathbb{R}^3$, the joint probability density function is denoted as:

$$p_{\mathbf{X}}(\mathbf{x}) \quad \text{where} \quad \mathbf{X} \in \mathbb{R}^3$$

If these are input observations, they are often referred to as "feature vectors."

1.1.5 Distinct Random Variables

We model samples as distinct random variables [1]:

$$\{x_1 \sim X_1, x_2 \sim X_2, \ldots, x_N \sim X_N\}$$

[1] **Why should we model these as distinct random variables? Aren't they all the same thing?**

Despite being identically distributed, distinct random variables allow for capturing dependencies and interactions between different samples.

1.2 Independent and Identically Distributed (i.i.d.) Random Variables

1.2.1 Independence of Random Variables

Independence refers to the idea that the values or outcomes of one observation in a dataset do not depend on or influence the values of any other observation.
For the set of random variables $X_1, X_2, \ldots X_n$, for the collection

$$\{x_1 \sim X_1, x_2 \sim X_2, \ldots, x_N \sim X_N\}$$

we often assume independence, for any subset of observations $\{X_{i1}, X_{i2}, \ldots, X_{ik}\}$ where $i_1, i_2, \ldots, i_k \in \{1, 2, \ldots, N\}$, the joint distribution factorises:

$$P(X_{i_1} = x_{i_1}, \ldots, X_{i_k} = x_{i_k}) = P(X_{i_1} = x_{i_1}) \cdot \ldots \cdot P(X_{i_k} = x_{i_k}) \tag{1.1}$$

The joint distribution of the subset is the product of the marginal distributions of the individual random variables.

Independence: The occurrence of any event does not affect the occurrence of others. This is a crucial assumption in many ML models that we will see the usefulness of as early as the next lecture.

Why do we model samples as distinct random variables?

1. By treating each sample as a distinct variable, we assume samples are i.i.d, allowing every sample to contribute independently to the likelihood of oberving the data given the model prarameters – so every sample provides unique information to estimate the parameters of the underlying distribution. If we treated all samples as a single random variable, we would lose the granularity of information, leading to a poorer estimate.
2. Treating samples as distinct allows us to study and model the relationships and dependencies between them.
3. The assumption of i.i.d follows many results in probability and statistics, such as the Central Limit Theorem, which states that the sum of a large number of i.i.d. random variables is approximately normally distributed. It is also an assumed requirement for a model to generalise. The assumption simplifies the analysis and derivation process, allowing us to use techniques like maximum likelihood estimation (MLE) and empirical risk minimization (ERM).

1.2.2 Identically Distributed Random Variables

The "identically distributed" part of i.i.d. refers to the idea that all random variables in the sample follow the same probability distribution. That is, they share the same probability density function (pdf) or probability mass function (pmf), depending on whether the data is continuous or discrete. If the set of random variables $\{X_1, X_2, \ldots, X_n\}$ are identically distributed, then:

$$f_{X_1}(x) = f_{X_2}(x) = \ldots = f_{X_n}(x) = f_X(x) \tag{1.2}$$

$$F_{X_1}(x) = F_{X_2}(x) = \ldots = F_{X_n}(x) = F_X(x) \tag{1.3}$$

This implies that the cumulative distribution function (CDF) is the same for all these random variables.

1.3 Statistical Modelling as Curve Fitting

Machine learning and statistics have significant overlap, and so most concepts studied in this module can be cast as statistical modelling. Consider our random variables:

$$\{x_1 \sim X_1, x_2 \sim X_2, \ldots, x_N \sim X_N\} \tag{1.4}$$

These random variables can be modelled to fit certain curves that represent the underlying data distribution.

1.3.1 Assumption about the Model

To proceed with statistical modelling, we make an assumption about the form of the model:

$$P_X(x) \approx \mathcal{N}(x; \theta) \quad \theta := (\mu, \sigma) \tag{1.5}$$

Here, $P_X(x)$ is approximated by a normal distribution $\mathcal{N}(x; \theta)$, where θ represents the parameters of the distribution, namely the mean μ and the standard deviation σ. This assumption simplifies the process of modelling the data.

1.3.2 Fitting the Model

The next step is to fit the model to the data. This involves finding the parameter values θ that maximize the probability of observing the given data.
Mathematically, this is expressed as:

$$\arg\max_{\theta} P(x_1, \ldots, x_N \mid \theta) \tag{1.6}$$

In other words, we adjust the parameters μ and σ so that the assumed model best fits the observed data. This process is known as maximum likelihood estimation (MLE).

1.4 Discrete Random Variables

1.4.1 Bernoulli and Multinoulli Distributions

Bernoulli Distribution: Outcome with two values (e.g., heads or tails).

- Parameter: θ.
- Probabilities: $P(X = 0) = 1 - \theta$, $P(X = 1) = \theta$.

Multinoulli Distribution: Describes a scenario with multiple possible outcomes, extending the Bernoulli distribution to more than two outcomes.

- Parameter: $\boldsymbol{\theta} \in \mathbb{R}^s$ where each θ_i represents the probability of the i-th outcome, and $\sum_{i=0}^{s-1} \theta_i = 1$.
- Probabilities: $P(X = i) = \theta_i$ for $i = 0, 1, \ldots, s-1$.

1.4.2 Binomial and Multinomial Distributions

Binomial Distribution: $X \sim \text{Bin}(n, \theta)$.

- Probability Mass Function (p.m.f):

$$\text{Bin}(k \mid n, \theta) := \binom{n}{k} \theta^k (1-\theta)^{n-k}$$

- Binomial Coefficient:

$$\binom{n}{k} = \frac{n!}{(n-k)!k!}$$

- Mean: $n\theta$.
- Variance: $n\theta(1-\theta)$.

Multinomial Distribution: Generalises the binomial distribution for more than two outcomes. It models the probabilities of counts among multiple categories in n independent trials.

- Parameters: n (number of trials) and $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$ where θ_i is the probability of the i-th category and $\sum_{i=1}^{K} \theta_i = 1$.
- Probability Mass Function (p.m.f):

$$\text{Mu}(\mathbf{x} \mid n, \boldsymbol{\theta}) := \binom{n}{x_1, x_2, \ldots, x_K} \prod_{j=1}^{K} \theta_j^{x_j}$$

  where $\mathbf{x} = (x_1, x_2, \ldots, x_K)$ represents the count of occurrences for each category.

- Multinomial Coefficient:

$$\binom{n}{x_1, x_2, \ldots, x_K} = \frac{n!}{x_1! x_2! \cdots x_K!}$$

- Mean for each category i: $\mathbb{E}[X_i] = n\theta_i$.
- Variance for each category i: $\text{Var}(X_i) = n\theta_i(1-\theta_i)$.
- Covariance between categories i and j: $\text{Cov}(X_i, X_j) = -n\theta_i\theta_j$.

1.4.3 Empirical Distribution

Empirical Distribution **Definition 1.4.1**

Based on observation or experience rather than theory or pure logic.

Empirical Distribution **Intuition 1.4.1**

Practically, we do not have access to an infinite amount of data, but we have instead a small fraction of it, a sample, to infer any insights from it. In the case of discrete random variables, we use probability mass functions, which is straightforward, but we are interested in probability density functions for continuous random variables, because to model the true distribution, we would need an infinite number of samples.

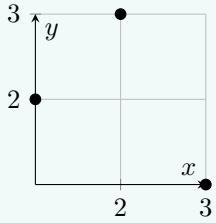
Thus, our goal is to approximate the true PDF from a given data set using finite samples. The transformation from discrete to continuous is done with the Dirac delta function.

A Dirac delta function is interestingly helpful because:

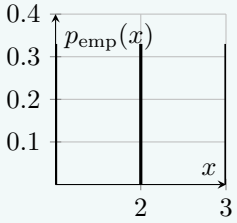
$$\int_{-\infty}^{\infty} \delta(x)\, dx = 1$$

We can then have multiple Dirac delta functions to represent the empirical distribution of a data set, but scaled down by a factor equivalent to the total number of data points to ensure the area under the curve is 1.

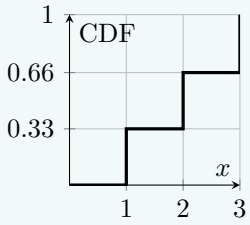
Plot 1: Data Points on a 2D Plane



Plot 2: Dirac Delta Functions



Plot 3: Cumulative Distribution Function (CDF)



Recommended Viewing

Lecture on Empirical Distribution

Empirical Statistics: When you compute statistics from a dataset, you're really computing statistics for its empirical distribution.

A dataset is, in essence, a distribution.

Reference 1.4.1

Empirical Distribution: Suppose we have a set of data samples

$$D = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$$

derived from a random variable X. We can approximate the distribution of X using a set of delta functions on these samples:

- For a given data set D, the empirical distribution $p_{\text{emp}}(x)$ is defined as:

$$p_{\text{emp}}(x) := \frac{1}{N} \sum_{i=1}^{N} \delta_{x_i}(x)$$

  where $\delta_{x_i}(x)$ is the Dirac measure centred at x_i.

- **Dirac Measure:** $\delta_{x_i}(x)$ is a function that is 1 if $x = x_i$ and 0 otherwise. Formally, it is defined as:

$$\delta_{x_i}(x) = \begin{cases} 1, & \text{if } x = x_i \\ 0, & \text{if } x \neq x_i \end{cases}$$

- Explanation: The empirical distribution assigns equal probability $\frac{1}{N}$ to each observed data point x_i. It is a discrete distribution that places mass only on the observed data points.
- In general, one can associate weights with each element of the empirical distribution, i.e., $p_{\text{emp}}(x) = \frac{1}{N}\sum_{i=1}^{N} w_i \delta_{x_i}(x)$ as long as each $0 \leq w_i \leq 1$ and $\sum_{i=1}^{N} w_i = 1$.

1.5 Continuous Random Variables

1.5.1 Gaussian (Normal) Distribution

Probability Density Function (p.d.f):

- Formula:
$$N(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$
- Mean: $\mu = \mathbb{E}[X]$.
- Variance: $\sigma^2 = \text{Var}[X]$.
- Standard Normal Distribution: $X \sim N(0, 1)$.
- Precision: $\lambda = \frac{1}{\sigma^2}$.

Cumulative Distribution Function (CDF):

- Formula:
$$\Phi(x; \mu, \sigma^2) = \int_{-\infty}^{x} N(z; \mu, \sigma^2)\, dz$$
- In terms of the error function (erf):
$$\Phi(x; \mu, \sigma^2) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{z}{\sqrt{2}}\right)\right]$$
where $z = \frac{x-\mu}{\sigma}$ and
$$\text{erf}(x) = \frac{1}{\sqrt{\pi}} \int_{0}^{x} \exp(-t^2)\, dt$$

1.6 Example Scene: Self-Driving Car

Note 1.6.1 Notation and Gotchas

The lecture use x_i to denote the i-th feature, and $x^{(i)}$ to denote the i-th data point. The slides however seem to denote x_i as the i-th data point. The notes follow the former convention.
Also, it is generally assumed that in a classification problem, the classes are distinct and mutually exclusive, so an image is either a dog or a cat (multi-class classfication) but not a combination of both (multi-label classification). The course focuses on the former convention.

Take a case of the self-driving car. There are several facets:

1. **Object Recognition:** identifying objects in the scene (classification)
   - An image is a 2D array of pixel values. For a colour image, each pixel has 3 values (RGB).

- **Input:**
$$x \in \mathbb{R}^{H \times W \times C} \tag{1.7}$$
where H is height, W is width, and C is the number of channels (e.g. 3 for RGB)
- **Output:** We would like to detect if something is a background, another vehicle, the ground level, or a pedestrian. Let's say there are m outcomes, making this a **classification problem**. Naively, let $y \in \mathbb{R}^m$. However, these numbers are just arbitrary i.e. raw scores. It would make more sense to refine the output to a probability distribution over the m classes:
$$y \in \Delta^m \quad \text{where} \sum_{i=1}^{m} y_i = 1 \tag{1.8}$$
Where the Δ^m is the m-simplex, where the sum of all elements is 1. This provides a measure of "confidence" in the prediction. Example: To choose between four classes: [Vehicle, Pedestrian, Road, Background], the output could be $[0.1, 0.7, 0.1, 0.1]$. This means the model is 70% confident that the object is a pedestrian.
- **Our objective is to learn a function:**
$$f^\theta : x^{(i)} \mapsto y^{(i)} \quad \text{where } \theta \text{ are model parameters}, x^{(i)} \in \mathbb{R}^{H \times W \times C}, y^{(i)} \in \Delta^m \tag{1.9}$$
A simplified example for f^θ is shown in Figure 1.3. Ideally, we want a separator that separates the input space into m distinct regions according to observed data.
- **Data Formalisation (1):** We can view our dataset as a set of N pairs where $x^{(i)}$ is an image and $y^{(i)}$ is the corresponding label. More neatly put,
$$\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N, \quad x^{(i)} \in \mathbb{R}^{H \times W \times C}, y^{(i)} \in \Delta^m \tag{1.10}$$
- **Data Formalisation (2):** We could also view the dataset as an empirical distribution over the data space like in Figure 1.4:
$$\mathcal{D} = \{x^{(i)}, y^{(i)}\}_{i=1}^N \sim p_{\text{data}}(x, y) \tag{1.11}$$

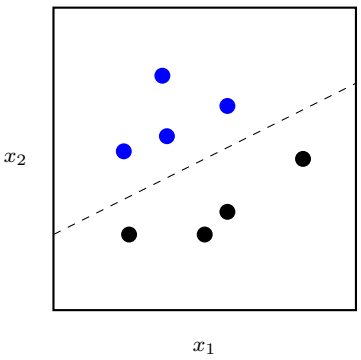


Figure 1.3: Simplified example for f^θ

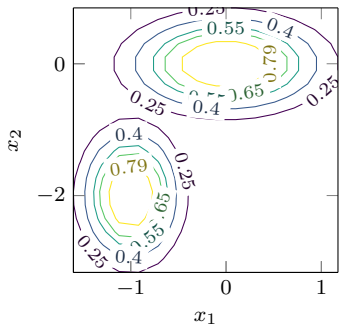


Figure 1.4: Viewing the dataset as an empirical distribution

2. **Route Planning:** We aim to predict the optimal path, where the prediction function f^θ takes spatial data and environmental conditions as input and outputs a route (either as a series of decisions or waypoints).
   - **Input:**
   $$x \in \mathbb{R}^n \tag{1.12}$$
   where x could represent a vector of features including road conditions, traffic density, and start/end coordinates.
   - **Output:** A route y, represented either as a classification over a set of discrete routes, or as continuous waypoints:
   $$f^\theta : x \mapsto y \tag{1.13}$$
   Here, $y \in \mathbb{R}^m$ for m possible waypoints (or classes, if we use a discrete route classification).

   The objective is to minimise the difference between predicted and actual routes, defined as a suitable loss function $\mathcal{L}(f^\theta(x), y)$.

3. **Speed Control:** In this problem, we predict the optimal driving speed, and f^θ models the relationship between environmental and driving conditions and speed.

- **Input:**
$$x \in \mathbb{R}^p \tag{1.14}$$
where x represents features such as current speed, distance to obstacles, road surface, and weather conditions.
- **Output:** A continuous speed value $y \in \mathbb{R}$:
$$f^\theta : x \mapsto y \tag{1.15}$$
where y is the predicted speed. This is a regression problem, so the goal is to minimise the squared error:
$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(f^\theta(x^{(i)}) - y^{(i)})^2 \tag{1.16}$$

4. **Steering Angle Prediction:** Steering angle prediction aims to output a continuous angle based on the driving conditions, making it a regression problem.
   - **Input:**
   $$x \in \mathbb{R}^q \tag{1.17}$$
   where x could represent lane position, vehicle surroundings, and road curvature.
   - **Output:** The predicted steering angle $y \in \mathbb{R}$:
   $$f^\theta : x \mapsto y \tag{1.18}$$
   This is also a regression task, and the loss function could again be the squared error:
   $$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(f^\theta(x^{(i)}) - y^{(i)})^2 \tag{1.19}$$

1.7 Mathematics of Linear Models

Linear models are fundamental in statistics and supervised machine learning. They can:

- Handle linear relationships.
- Be augmented with kernels or basis functions to model non-linear relationships.
- Provide analytical tractability for studying concepts like convergence, probabilistic modelling, and overfitting.

This section introduces the linear regression model.

1.7.1 Linear Regression Model

Linear regression involves performing regression with a linear model in a supervised setting. Given a dataset $\mathcal{D}$ consisting of input-output pairs $(\boldsymbol{x}^{(i)}, y^{(i)})$ for $i = 1, \ldots, N$:

- $\boldsymbol{x} \in \mathbb{R}^n$ represents the input features.
- $y \in \mathbb{R}$ represents the output or target variable.
- Assume a domain $\mathbb{R}^n$ and a one-dimensional co-domain.
- Model: $f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\theta}$.
- Noise: $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

The full model is:
$$\hat{y}^{(i)} = \boldsymbol{x}^{(i)\top}\boldsymbol{\theta} + \epsilon$$

The objective is to find $\boldsymbol{\theta}$ such that $\hat{y}^{(i)} \approx y^{(i)}$. In other words, we want to find a set of parameters $\boldsymbol{\theta}$ that best explains the relationship between the input features $\boldsymbol{x}$ and the target variable y. This means minimising the difference between the predicted values $\hat{y}$ and the actual values y.

Conventions and Notations:

- Vectors $\boldsymbol{x} \in \mathbb{R}^n$ are column vectors, written as $n \times 1$ matrices.
- $\boldsymbol{x}^\top$ (x transpose) swaps rows and columns of $\boldsymbol{x}$, resulting in a $1 \times n$ row vector.

Lecture Reading **Reference 1.7.1**

Chapter 1-2 of Kevin Murphy's *Machine Learning: A Probabilistic Perspective*. Chapter 1.4 was not covered, but may be of interest

2
Towards Learning in Linear Models

2.1 Advanced Problem Formulations (Slides)

The below are a list of major subject areas in NeurIPS [1]:

[1] As of 2024, when the slides were created.

- Supervised, Unsupervised or Semi-Supervised Learning:
  - Few-shot Learning
  - Transfer Learning
- Unsupervised Learning:
  - Density Estimation
  - Clustering
  - Dimensionality Reduction

2.1.1 Few-shot Learning

In supervised learning, let us define the notation:
Let our feature vector (which is our input space) be defined by

$$x \in \mathbb{R}^n \tag{2.1}$$

Let our output labels be defined by

$$y \in \mathbb{R}^m \tag{2.2}$$

Assume our dataset is i.i.d distributed and $K >> n$, it is defined as

$$\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{K} \tag{2.3}$$

Our learning objective is to find the minimum error from our loss function $\mathcal{L}$.

$$\min_{\theta} \left[\mathcal{L}(f^{\theta}(x), y)\right] \tag{2.4}$$

In the case of **few-shot learning**, a technique in ML where a model learns to perform a task proficiently with only a limited amount of training data. We do not have the luxury of $K \gg n$ and so our model must learn to generalise well, quickly.

- When $K = n$:
  - **n-shot learning**: Trains with n examples per class.
  - Example: With $n = 3$, learn to recognise animals like cats, dogs, and birds from three images each.
- When $K = 0$:
  - **zero-shot learning**:[1] Infers classes with no examples, using descriptions.
  - Example: Identify an animal as a mammal based on descriptions of mammals being warm-blooded with hair.
- When $K = nc$:
  - **n-way c-shot learning**: Trains with c examples from each of n classes.
  - Example: In a 5-way 2-shot scenario, classify fruits like apples, oranges, bananas, grapes, pineapples from two images each.

[1] Very popular with Large Language Models (LLMs) and also known as zero-shot prompting in this case.

2.1.2 Transfer Learning

Transfer learning is a machine learning technique where a model trained on one task is re-purposed on a second related task. Let us define two datasets, $\mathcal{D}$ and $\mathcal{D}'$:

$$\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N} \quad \text{where } x^{(i)} \in \mathbb{R}^n, y^{(i)} \in \mathbb{R}^d \tag{2.5}$$

$$\mathcal{D}' = \{(x'^{(i)}, y'^{(i)})\}_{i=1}^{N'} \quad \text{where } x'^{(i)} \in \mathbb{R}^n, y'^{(i)} \in \mathbb{R}^d \tag{2.6}$$

We then define the functions used in transfer learning :[2]

[2] Bear with the difference in notation here: the notes use ϕ to represent the feature extractor instead of g.

$$g^{\omega} : \mathbb{R}^n \to \mathbb{R}^l \tag{2.7}$$

$$h^{\kappa} : \mathbb{R}^l \to \mathbb{R}^d \tag{2.8}$$

$$f^{\theta} : \mathbb{R}^l \to \mathbb{R}^m \tag{2.9}$$

The function g^{ω} represents a feature extractor that maps input features from $\mathbb{R}^n$ to a latent space $\mathbb{R}^l$. The function h^{κ} is a task-specific classifier or regressor for the old task, mapping the latent features to outputs in $\mathbb{R}^d$. The function f^{θ} is a classifier or regressor for the new task, also mapping the latent features to outputs in $\mathbb{R}^m$.

The learning process in transfer learning typically involves two main steps:

1. Pre-training on the old dataset $\mathcal{D}$:

$$\min_{\omega,\kappa} \mathbb{E}_{\mathcal{D}}[L(y, h^{\kappa}(g^{\omega}(x)))] \tag{2.10}$$

   This step involves optimising the parameters ω and κ to minimise the expected loss L on the old dataset $\mathcal{D}$, effectively training the feature extractor g^{ω} and the old task classifier h^{κ}.

2. Fine-tuning on the new dataset $\mathcal{D}'$:

$$\min_{\omega,\theta} \mathbb{E}_{\mathcal{D}'}[L(y', f^{\theta}(g^{\omega}(x')))] \tag{2.11}$$

   In this step, the feature extractor g^{ω} is further optimised along with the new task classifier f^{θ} to minimise the loss on the new dataset $\mathcal{D}'$. The parameters ω are fine-tuned to adapt to the new task, leveraging the feature extraction capabilities learned from the old task.

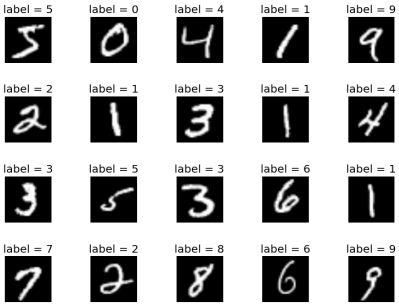
2.1.3 Density Estimation, Clustering and Dimensionality Reduction

Assume we have an unsupervised learning setup, with $x \in \mathbb{R}^n$ and our dataset $\mathcal{D} = \{x_1, x_2, \ldots, x_K\}$ where data is i.i.d and $K \gg n$.
Notes:

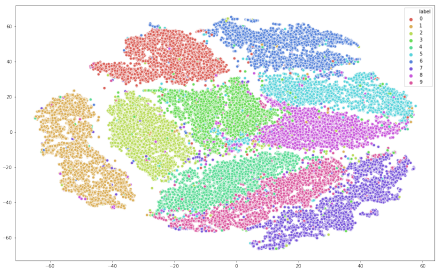
1. **Density Estimation** is to estimate the probability density function (pdf) of the data distribution.

2. **Clustering** is to find a way to group data points as clusters. We typically first estimate how many clusters exists, then try to fit assign points to the clusters.

3. **Dimensionality Reduction** involves finding latent factors and/or associations in the data. Even though though the spacec of the data is large, the effective dimensionality of the problem is small. For example, we can visualise the MNIST dataset of handdrawn digits from 1-10 in a 2D space with t-Distributed Stochastic Neighbor Embedding (t-SNE) in Figure 2.1.



(a) MNIST dataset



(b) t-SNE visualisation on MNIST dataset

Figure 2.1: MNIST dataset (dimension 10) and its corresponding t-SNE visualisation

2.2 Supervised Learning

Supervised learning assumes that we have access to input-output pairs of datapoints, $(\mathbf{x}, \mathbf{y})$, with $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$, forming our dataset[3] $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}$.

[3] The slides replaced K with N. Notation!

2.3 Linear Regression Model

Assuming a domain of $\mathbb{R}^n$ and a one-dimensional co-domain, we can write our model as $f(\mathbf{x}) = \mathbf{x}^\top \theta$. Thus we have:

$$\hat{y}^{(i)} = \mathbf{x}^{(i)\top} \theta$$

The goal of learning is to find θ such that $\hat{y}^{(i)} \approx y^{(i)}$.

Linear Transformation **Definition 2.3.1**

A linear transformation between two vector spaces V and W is a map

$$T : V \to W$$

such that:

- $T(v_1 + v_2) = T(v_1) + T(v_2) \quad \forall v_1, v_2 \in V$
- $T(\alpha v) = \alpha T(v) \quad \forall v \in V$ and scalar α

You may recall linear models written as affine transformations: $y = mx + b$, where b is the bias or constant term – this makes the model affine and not linear. Linear models refers to the relationship between model parameters and predictions via a linear transformation.

Affine Transformations

Affine transformations are more general than linear transformations, because they include not only scaling and rotation, but also translations.

Definition 2.3.2

Now we are faced with something interesting: $T(\mathbf{0}) = \mathbf{0}$. According to our affine transformation $f(x) = mx + b$ where $m, b \in \mathbb{R}$, we have $f(0x) = b \neq 0f(x)$, which doesn't allow us to have a bias term b. This is easily fixed with a straightforward modification to capture the affine transformation $f(x) = mx + b$: we just add a feature to the input vector x that is always equal to 1, then the corresponding weight for this feature becomes the bias. Introduce:

$$\phi(x) : \mathbf{R} \Rightarrow \mathbf{R}^2 \quad \text{such that } \phi(x) = \begin{bmatrix} 1 \\ x \end{bmatrix} \tag{2.12}$$

We then need parameter vector $\theta = \begin{bmatrix} b \\ m \end{bmatrix}$.

We then have the model

$$\hat{y} = \phi(x)^\top \theta \Rightarrow \begin{bmatrix} 1 & x \end{bmatrix} \begin{bmatrix} b \\ m \end{bmatrix} = b + mx \tag{2.13}$$

2.4 Basis Expansion

As seen in the previous section, a model that was once restricted to lines through the origin ahs been expanded to fit the affine transformation with the aid of Basis Expansion. We can also more generally utilise it to model non-linear relationships.

The key idea of basis expansion is to expand a one-dimensional feature into many dimensions, and use non-linear functions to increase the expressiveness of the model.

2.4.1 Example Polynomial Basis Expansion

A one dimensional domain $x \in \mathbb{R}$ and a one dimensional co-domain $y \in \mathbb{R}$ is assumed. Our model is $\hat{y} = \phi(x)^\top \theta$.

We choose the basis:

$$\phi(x) = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix} \quad \phi(x) : \mathbb{R}^1 \to \mathbb{R}^3 \tag{2.14}$$

and weights:

$$\theta \in \mathbb{R}^3 \tag{2.15}$$

We finally have the fully expanded function:

We use the dot product instead of transposing, for clarity of notation.

$$\hat{y} = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix} \cdot \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} = \theta_0 + \theta_1 x + \theta_2 x^2 \tag{2.16}$$

This example model is a quadratic polynomial.

2.4.2 Another Example Polynomial Basis Expansion

Again, given our model is $\hat{y} = \phi(\mathbf{x})^\top \theta$ where $y \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^2$. We can use the basis

$$\phi(\mathbf{x}) \Rightarrow \phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1 x_2 \\ x_1^2 \\ x_2^2 \end{bmatrix} \quad \phi(x) : \mathbb{R}^2 \to \mathbb{R}^6 \tag{2.17}$$

and with a new set of corresponding weights, we have:

$$\hat{y} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1 x_2 \\ x_1^2 \\ x_2^2 \end{bmatrix} \cdot \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{bmatrix} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 \quad \theta \in \mathbb{R}^6 \tag{2.18}$$

2.5 Radial Basis Function Kernel

Polynomial basis expansion is just a single flabour of basis expansion. Another widely-used form of basis is the kernel basis expansion. One popular example is

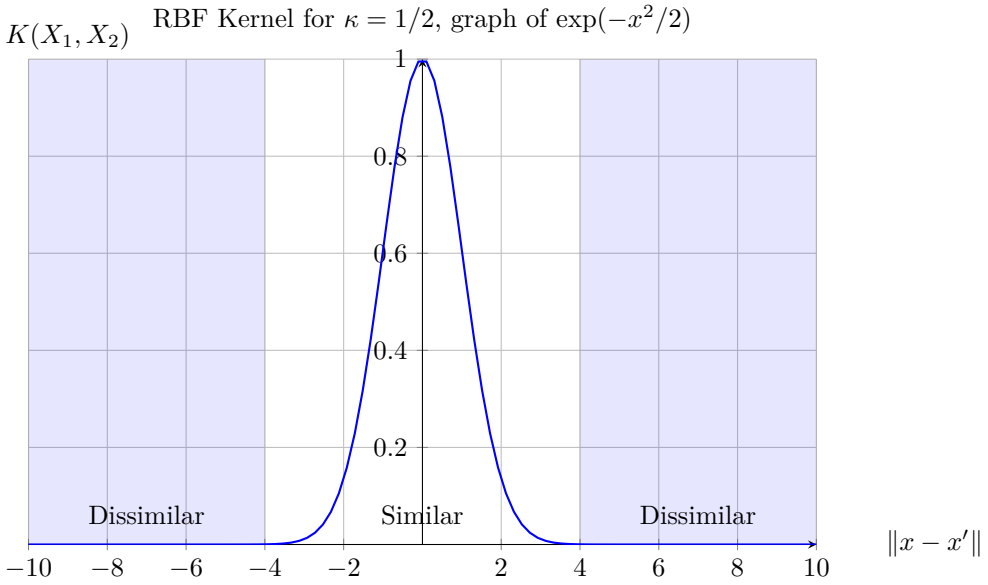
the radial basis function kernel (RBF kernel), which is a generalisation of the polynomial basis expansion.

It takes in a fixed parameter $\gamma > 0$, defined as

$$\kappa(x, x') = \exp(-\gamma||x - x'||^2) \tag{2.19}$$

where $||x - x'||^2$ is the squared Euclidean distance (or more appropriately, the L_2 Norm) between x and x'. In practice, one picks fixed centres x and the basis expansion computes the expanded feature set w.r.t the distance to these centres.[4]

- It is easy to see that the kernel basis expansion $\kappa(x, x')$ has a minimum value of 0. It takes a maximum value of 1 when $x = x'$.
- When two points x and x' are far apart, the kernel value is closer to 0, and when they are closer together, the kernel value is closer to 1.
- It is quite akin to a simiarity score, and that a smaller value of γ leads to larger similarity scores (see Figure 2.2).
- It is also symmetric, $\kappa(x, x') = \kappa(x', x)$, and is always positive.



The RBF kernel is actually a special case of the polynomial basis expansion, where

2.5.1 Example Radial Basis Function Kernel

Given a one-dimensional domain and a one-dimensional co-domain, we have the model $\hat{y} = \phi(x)^\top \theta$.

$$\phi(x) = \exp(-\gamma||x - x'||^2) \quad \phi(x) : \mathbb{R}^1 \to \mathbb{R}^1 \tag{2.20}$$

and with a new set of corresponding weights, we have:

$$\hat{y} = \exp(-\gamma||x - x'||^2) \cdot \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \theta_0 \exp(-\gamma||x - x'||^2) + \theta_1 \tag{2.21}$$

[4] There is more nuance to this: you may have noticed that kernel functions take in two points, unlike the polynomial basis expansion ϕ taking in a single point. This is because for each of the n points, we compute pairwise similarities with a point's other points, and then construct a feature vector for each point, where each point x is transformed into a n-dimensional feature vector where each dimension represents the similarity between x and one of the n points.

For example, given a dataset with three points x_1, x_2, x_3, and a radial basis function $\kappa(x, x')$, the RBF kernel matrix might look like this:

$$K = \begin{bmatrix} \kappa(x_1, x_1) & \kappa(x_1, x_2) & \kappa(x_1, x_3) \\ \kappa(x_2, x_1) & \kappa(x_2, x_2) & \kappa(x_2, x_3) \\ \kappa(x_3, x_1) & \kappa(x_3, x_2) & \kappa(x_3, x_3) \end{bmatrix}$$

where the feature vector for x_1 as an example would be

$$\begin{bmatrix} \kappa(x_1, x_1) & \kappa(x_1, x_2) & \kappa(x_1, x_3) \end{bmatrix}$$

This is related to the use of the kernel trick.

Figure 2.2: RBF Kernel for $\kappa = 1/2$, graph of $\exp(-x^2/2)$. Areas of dissimilarity are subjectively noted where the kernel values are negligible.

2.6 Linear Algebra

2.6.1 Vectors

A vector is defined by a list of numbers. It is most useful to geometrically interpret vectors as points in space.

$$\mathbf{x} := \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{2.22}$$

We have several vector operations:

- **Scalar Multiplication**:

$$\alpha\mathbf{x} = \begin{bmatrix} \alpha x_1 \\ \alpha x_2 \\ \vdots \\ \alpha x_n \end{bmatrix} \tag{2.23}$$

- **Vector Addition**:

$$\mathbf{x} + \mathbf{y} = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_n + y_n \end{bmatrix} \tag{2.24}$$

- **Dot Product**:

$$\mathbf{x}^\top \mathbf{y} = \sum_{i=1}^{n} x_i y_i \tag{2.25}$$

2.6.2 Matrices

Matrices define linear transforms. Geometrically, it is best to interpret them as transformations whose columns are the new basis vectors.

$$\mathbf{A} := \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,n} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m,1} & A_{m,2} & \cdots & A_{m,n} \end{bmatrix} \quad \mathbf{A} \in \mathbb{R}^{m \times n} \tag{2.26}$$

Non-square matrices can be thought of as transformations from $\mathbb{R}^m$ dimentional space to $\mathbb{R}^n$ dimensional space.

2.6.3 Matrix Operations

We define several matrix operations as follows:

- **Scalar Multiplication**:

$$cA = cA_{i,j} \tag{2.27}$$

- **Hadamard Product (Element-wise Multiplication)**:

$$A \circ B = A_{i,j} B_{i,j} \tag{2.28}$$

- **Matrix Subtraction**:

$$A - B = A_{i,j} - B_{i,j} \tag{2.29}$$

- **Matrix Product**:

$$C_{i,j} = \sum_{k} A_{i,k} B_{k,j} \tag{2.30}$$

- **Trace of a Matrix**:

$$\mathrm{Tr}(A) = \sum_i A_{i,i} \tag{2.31}$$

Dimensional Notation:

$$A \in \mathbb{R}^{n\times m} : \mathbb{R}^n \to \mathbb{R}^m \tag{2.32}$$

$$B \in \mathbb{R}^{m\times k} : \mathbb{R}^m \to \mathbb{R}^k \tag{2.33}$$

$$AB \in \mathbb{R}^{n\times k} : \mathbb{R}^n \to \mathbb{R}^k \tag{2.34}$$

Note 2.6.1 Matrix Multiplication as Dot Products of the Row and Column

This matrix multiplication is equivalent to the dot product between row i of matrix A and column j of matrix B:

$$\mathbf{x}^\top \mathbf{y} = \sum_{i=1}^{n} x_i y_i \tag{2.35}$$

2.6.4 Einstein/Pythonic Index Notation

Most readers will have covered a lot of the linear algebra basics before, if so, this is the most important section to read!

Refer to this video on Einstein summation convention for more information.

As reasoning about matrix and vector products can sometimes be cumbersome, it is often useful to write out the operations we perform in index notation. The matrix product $C = AB$ can be written as: $C_{i,j} = \sum_k A_{i,k} B_{k,j}$. It can also be useful to adopt a more "pythonic" index notation where we consider the system $Ax = b$ and write the first entry of the vector b as:

$$A_{1,:}x = b_1 \tag{2.36}$$

which indicates that the first value of the result is simply the dot product of the first row of matrix A with the vector x.
Then, we can write the entire system as:

$$C_{i,j} = \sum A_{i,:} B_{:,j} \tag{2.37}$$

Generally in tensor calculus, a lot of expressions involve summing over particular indices.

$$\sum_{i=1}^{3} a_i x_i = a_1 x_1 + a_2 x_2 + a_3 x_3 \tag{2.38}$$

In **Einstein notation**, we can write this as:

$$a_i x_i \quad (i = 1, 2, 3) \tag{2.39}$$

Fun fact

Yes, this was introduced by Albert Einstein in 1916!

Non-Examinable **2.6.1**

This brings us to our first rule (and several others):

Rule 1: Any twice-repeated index in a single term is implicitly summed over. Typically, this is from index 1 to 3 because most calculations are done in 3D space. For example, if we had:

$$a_{ij} b_j = a_{i1} b_1 + a_{i2} b_2 + a_{i3} b_3 \tag{2.40}$$

We can then express this in the Einstein notation as:

$$a_{ij} b_j = a_{i\alpha} b_\alpha \quad \alpha \in \{1, 2, 3\} \tag{2.41}$$

This will begin to make more sense as we come up with a better way to describe indices that appear once, and twice-repeated indices.

Rule 2: The definitions of indices:

- We let j be the dummy index, because it is repeated only twice. One can thus replace j with any other index or letter, it is just a placeholder (thus called a dummy index). Although more rigorously:
  - One can replace any dummy index with a letter/index that is not already used in the expression.
  - This letter must be over the same range as the original dummy index, so in the case of replacing j, it must be over the range 1 to 3.

$$a_{ij}b_j = a_{i1}b_1 + a_{i2}b_2 + a_{i3}b_3 \tag{2.42}$$
$$= a_{i\alpha}b_\alpha \quad \alpha \in \{1,2,3\} \tag{2.43}$$

- i is the free index, which can take on any value that j takes on, but it is not summed over and can only take on one value at a time $i \in \{1,2,3\}$.
- The free index occurs only **once** in the expression and **cannot be replaced by another free index,**

$$a_{ij}b_j \neq a_{kj}b_j \tag{2.44}$$

- To help avoid confusion, one tip is to use roman letters (i, j, k) for free indices, and greek letters (λ, μ, ρ) for dummy indices.

Rule 3: No index may occur 3 or more times in a given term.

- $a_{ij}b_{ij}$ ✓
- $a_{ii}b_{ij}$ ×
- $a_{ij}b_{ij}$ ×
- $a_{ij}b_j + a_{ji}b_j$ ✓

In the last example, we are adding **multiple terms**, so the index occurence rule only applies by term. So j is a dummy index for both terms since it occurs twice (per term).

Rule 4: In an equation involving Einstein notation, the free indices on the left-hand side must match the right-hand side.

- $x_i = a_{ij}b_j$ ✓
  - i is a free index on both the LHS and RHS.
- $a_i = A_{xi}B_{xk}x_j + C_{ik}u_k$ ✓
  - i is the free index on both the LHS and RHS.
- $x_i = A_{ij}$ ×
  - i is a free index on both the LHS and RHS, but j is a free index on the RHS that is not on the LHS.
- $x_j = A_{ik}u_k$ ×
  - LHS free index: j.
  - RHS free index: i.
- $x_i = A_{ik}u_k + c_j$ ×
  - LHS free index: i.
  - RHS free indices: i and j.

Relating Einstein Notation to Pythonic Notation: We had:

$$C_{i,j} = \sum_k A_{i,k} B_{k,j} \tag{2.45}$$

Since k is our dummy variable (it is summed over and doesn't appear in the final result), we can replace it with colon (:) to indicate we are working with all elements along that dimension. So we have

$$C_{i,j} = A_{i,:} B_{:,j} \tag{2.46}$$

In Python, this is:

```
C[i, j] = np.dot(A[i, :], B[:, j])
```

2.6.5 Matrix Properties: Linear Dependence

Two vectors are said to be **linearly dependent** if one is a scalar multiple of the other. That is, for vectors $\mathbf{a}$ and $\mathbf{b}$, we have:

$$\mathbf{a} = \alpha \mathbf{b} \tag{2.47}$$

where $\alpha \in \mathbb{R}$ is a scalar.
A set of vectors $\{\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \ldots, \mathbf{a}^{(n)}\}$ is linearly dependent if there exist non-zero real weights $v^{(i)}$ such that their weighted sum results in the zero vector:

$$v^{(1)}\mathbf{a}^{(1)} + \cdots + v^{(n)}\mathbf{a}^{(n)} = 0 \tag{2.48}$$

This implies that at least one vector in the set can be written as a linear combination of the others:

$$\mathbf{a}^{(1)} = -\frac{v^{(2)}}{v^{(1)}}\mathbf{a}^{(2)} - \cdots - \frac{v^{(n)}}{v^{(1)}}\mathbf{a}^{(n)} \tag{2.49}$$

2.6.6 Span of Vectors

The **span** of a set of vectors $\{\mathbf{v}^{(i)}\}_{i=1}^K$ is the set of all possible linear combinations of these vectors. Formally, the span is given by:

$$\text{Span}(V) = \left\{ \sum_{i=1}^K \lambda^{(i)} \mathbf{v}^{(i)} \mid \lambda^{(i)} \in \mathbb{R} \right\} \tag{2.50}$$

Here, the coefficients $\lambda^{(i)}$ are real numbers, and the span defines the vector subspace formed by these linear combinations.

2.6.7 From Span to Rank

The **rank** of a matrix A, formed by stacking column vectors $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(K)}$, is the dimension of the vector space spanned by these vectors. If:

$$A = \begin{bmatrix} \mathbf{a}^{(1)\top} & \cdots & \mathbf{a}^{(K)\top} \end{bmatrix} \tag{2.51}$$

Then:

$$\text{Rank}(A) = \text{Span}\left(\mathbf{a}^{(1)\top}, \ldots, \mathbf{a}^{(K)\top}\right) \tag{2.52}$$

A matrix is said to be **full rank** if the rank of the matrix equals the number of its rows or columns (i.e., if an $n \times n$ matrix has rank n).

2.6.8 Eigen Decomposition

Given a matrix $A \in \mathbb{R}^{n \times n}$, an **eigenvector** $\mathbf{v}$ and an **eigenvalue** λ satisfy the following equation:

$$A\mathbf{v} = \lambda\mathbf{v} \tag{2.53}$$

Note that any scalar multiple of $\mathbf{v}$ is also an eigenvector. To standardise, we typically normalise the eigenvector to have unit length:

$$\|\mathbf{v}\|_2 = 1 \tag{2.54}$$

This ensures that the eigenvector is unique up to a scalar factor.

2.6.9 Eigen Decomposition of a Matrix

The **eigen decomposition** of a matrix $A \in \mathbb{R}^{n \times n}$ expresses the matrix in terms of its eigenvalues and eigenvectors. Formally, this decomposition is given by:

$$A = Q\Lambda Q^{-1} \tag{2.55}$$

where:

- Q is the matrix whose columns are the eigenvectors of A.
- Λ is a diagonal matrix with the eigenvalues λ_i of A on the diagonal.

The eigen decomposition allows us to express A as a product of its eigenvectors and eigenvalues, enabling many applications in linear algebra, such as simplifying powers of matrices or solving systems of linear equations.

2.6.10 Norms

A **norm** is a function that assigns a non-negative length or size to a vector. Norms are widely used in mathematics to measure distances or lengths in vector spaces. Below are some commonly used norms:

- **Euclidean Norm (2-norm)**: Also known as the straight-line distance, the Euclidean norm is given by:

$$\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2} \tag{2.56}$$

- **p-norm**: The p-norm generalises the Euclidean norm to other values of p:

$$\|\mathbf{x}\|_p = \left(\sum_i |x_i|^p\right)^{1/p} \tag{2.57}$$

  Special cases of the p-norm include:

  - The 2-norm (Euclidean norm) when $p = 2$.
  - The 1-norm, which represents the Manhattan distance, when $p = 1$.

- **Zero Norm (Hamming Distance)**: The 0-norm counts the number of non-zero elements in a vector:

$$\|\mathbf{x}\|_0 = \sum_i \mathbb{I}(x_i \neq 0) \tag{2.58}$$

  where $\mathbb{I}(\cdot)$ is the indicator function, which returns 1 if the condition is true and 0 otherwise.

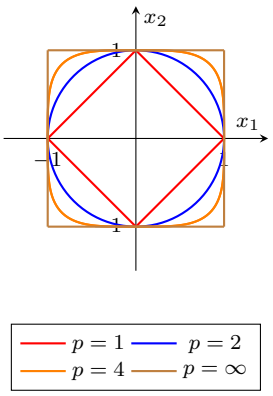


Figure 2.3: Different norms in 2D space.

- **Infinity Norm**: Also called the *supremum norm*, the infinity norm is defined as the largest absolute value among the vector components:

$$\|\mathbf{x}\|_\infty = \sup_n |x_n| \tag{2.59}$$

- **Mahalanobis Distance**: The Mahalanobis distance takes into account the correlations between variables in a dataset:

$$d_S(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T S(\mathbf{x} - \mathbf{y}) \tag{2.60}$$

  where S is a positive semi-definite matrix, typically the covariance matrix of the data.

Norms are typically used as a way to measure distances and magnitudes.

Theorem 2.6.1 Geometric Interpretation of the Mahalanobis Distance

Given a positive semi-definite matrix $S \in \mathbb{R}^{m\times m}$, a feature vector $\mathbf{x}' \in \mathbb{R}^m$, and a similarity threshold δ, all vectors $\mathbf{x}'' \in \mathbb{R}^n$ satisfying $d_S(\mathbf{x}', \mathbf{x}'') \leq \delta$ are contained within the axis-aligned orthotope:

$$[\mathbf{x}' - \delta\sqrt{\mathbf{d}}, \mathbf{x}' + \delta\sqrt{\mathbf{d}}]$$

where $\mathbf{d} = \text{diag}(S)$, the vector containing the elements along the diagonal of S.

2.6.11 Equivalence of Inner Product and Euclidean Norm

We aim to prove the equivalence between the quadratic form and the squared Euclidean norm:

$$\text{Claim: } (\theta^T\mathbf{X} - \mathbf{y})^T(\theta^T\mathbf{X} - \mathbf{y}) = \|\theta^T\mathbf{X} - \mathbf{y}\|_2^2 \tag{2.61}$$

Left-hand side (LHS): The quadratic form on the LHS can be expanded as:

$$(\theta^T\mathbf{X} - \mathbf{y})^T(\theta^T\mathbf{X} - \mathbf{y}) = \left(\left(\sum_i \theta_i^\top \mathbf{X}_{i,j}\right) - \mathbf{y}_j\right)^\top \left(\left(\sum_i \theta_i^\top \mathbf{X}_{i,j}\right) - \mathbf{y}_j\right) \tag{2.62}$$

$$= \sum_j \left(\left(\sum_i \theta_i^\top \mathbf{X}_{i,j}\right) - \mathbf{y}_j\right)^\top \left(\left(\sum_i \theta_i^\top \mathbf{X}_{i,j}\right) - \mathbf{y}_j\right) \tag{2.63}$$

$$= \sum_j \left(\left(\sum_i \theta_i^\top \mathbf{X}_{i,j}\right) - \mathbf{y}_j\right)^2 \tag{2.64}$$

Right-hand side (RHS): The squared Euclidean norm is defined as:

$$\|\theta^T\mathbf{X} - \mathbf{y}\|_2^2 \tag{2.65}$$

We know that the Euclidean norm is given by:

$$\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2} \tag{2.66}$$

Therefore, we expand into:

$$\|\theta^T\mathbf{X} - \mathbf{y}\|_2^2 = \left(\sqrt{\sum_j \left(\sum_i \theta_i^T \mathbf{X}_{i,j} - \mathbf{y}_j\right)^2}\right)^2 \tag{2.67}$$

$$= \sum_j \left(\sum_i \theta_i^T \mathbf{X}_{i,j} - \mathbf{y}_j\right)^2 \tag{2.68}$$

Conclusion: Both the LHS and RHS are equivalent, as they both expand into the same form:

$$\sum_j \left(\sum_i \theta_i^T \mathbf{X}_{i,j} - \mathbf{y}_j \right)^2 \tag{2.69}$$

This shows that the inner product expression for $(\theta^T \mathbf{X} - \mathbf{y})^T (\theta^T \mathbf{X} - \mathbf{y})$ is equivalent to the squared Euclidean norm $\|\theta^T \mathbf{X} - \mathbf{y}\|_2^2$, confirming the claim.

To finish it off in index notation, this is written as:

$$\theta_i \mathbf{X}_{i,j} \tag{2.70}$$

2.7 Revisiting Calculus

Recall from your previous studies that the derivative of a real-valued function $f(x)$ is defined as the limit of the difference quotient:

$$f'(x) = \frac{df}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \tag{2.71}$$

This limit definition is important to understanding how functions change locally around a particular point. While differentiation can sometimes be difficult to compute directly, this formula also provides a method for approximating the derivative numerically. For any interested reader, the method of zeroth-order optimisation can be explored for more computational approaches.
Several useful derivatives of common functions include:

$$\begin{aligned}
&f(x) = x^n, && f'(x) = nx^{n-1} \\
&f(x) = \sin(x), && f'(x) = \cos(x) \\
&f(x) = \tanh(x), && f'(x) = 1 - \tanh^2(x) \\
&f(x) = \exp(x), && f'(x) = \exp(x) \\
&f(x) = \log(x), && f'(x) = \frac{1}{x}
\end{aligned}$$

It is important to recall the key rules of differentiation that allow us to easily compute derivatives of more complex expressions. These rules are essential for differentiating composite functions and will be useful throughout the course.

2.7.1 Sum Rule

The derivative of the sum of two functions is given by:

$$(f(x) + g(x))' = f'(x) + g'(x) = \frac{df(x)}{dx} + \frac{dg(x)}{dx} \tag{2.72}$$

2.7.2 Product Rule

The derivative of the product of two functions is given by:

$$(f(x)g(x))' = f'(x)g(x) + f(x)g'(x) = \frac{df(x)}{dx} g(x) + f(x) \frac{dg(x)}{dx} \tag{2.73}$$

2.7.3 Chain Rule

The derivative of the composition of two functions is given by:

$$(g \circ f)'(x) = (g(f(x)))' = g'(f(x)) f'(x) = \frac{dg(f(x))}{df} \frac{df(x)}{dx} \tag{2.74}$$

2.7.4 Quotient Rule

The derivative of the quotient of two functions is given by:

$$\left(\frac{f(x)}{g(x)}\right)' = \frac{f'(x)g(x) - f(x)g'(x)}{(g(x))^2} = \frac{\frac{df(x)}{dx}g(x) - f(x)\frac{dg(x)}{dx}}{(g(x))^2} \tag{2.75}$$

These rules will be useful as we move forward and encounter more complex functions to differentiate.

2.8 Gradients and Partial Derivatives

The derivative can be generalised for functions $f : \mathbb{R}^n \to \mathbb{R}$, where $f(x)$ depends on multiple variables $x_1, x_2, \ldots, x_n$. We compute the **partial derivative** with respect to each variable by holding the others constant:

$$\frac{\partial f}{\partial x_1} = \lim_{h\to 0} \frac{f(x_1 + h, x_2, \ldots, x_n) - f(x_1, x_2, \ldots, x_n)}{h} \tag{2.76}$$

We collect these partial derivatives into the **gradient vector**:

$$\nabla f = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right] \tag{2.77}$$

When dealing with vector-valued functions $f : \mathbb{R}^n \to \mathbb{R}^m$, the gradient generalises to the **Jacobian matrix** $\nabla f \in \mathbb{R}^{m\times n}$:

$$\nabla f = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{2.78}$$

For functions $f : \mathbb{R}^{n\times m} \to \mathbb{R}^{k\times l}$, the gradient forms a tensor of shape $(k \times l) \times (n \times m)$, where each element represents the partial derivative of an output with respect to an input variable.

2.8.1 Vector Calculus Identities to Remember

Below are some key vector calculus identities that will prove useful in various contexts:

$$\begin{aligned}
\frac{\partial \boldsymbol{x}^\top \boldsymbol{a}}{\partial \boldsymbol{x}} &= \boldsymbol{a}^\top \\
\frac{\partial}{\partial \boldsymbol{X}} f(\boldsymbol{X})^\top &= \left(\frac{\partial f(\boldsymbol{X})}{\partial \boldsymbol{X}}\right)^\top \\
\frac{\partial}{\partial \boldsymbol{X}} \mathrm{tr}(f(\boldsymbol{X})) &= \mathrm{tr}\left(\frac{\partial f(\boldsymbol{X})}{\partial \boldsymbol{X}}\right) \\
\frac{\partial}{\partial \boldsymbol{X}} f(\boldsymbol{X})^{-1} &= -f(\boldsymbol{X})^{-1}\frac{\partial f(\boldsymbol{X})}{\partial \boldsymbol{X}} f(\boldsymbol{X})^{-1} \\
\frac{\partial \boldsymbol{a}^\top \boldsymbol{X} \boldsymbol{b}}{\partial \boldsymbol{X}} &= \boldsymbol{a}\boldsymbol{b}^\top \\
\frac{\partial}{\partial s}(\boldsymbol{x} - \boldsymbol{A}s)^\top \boldsymbol{W}(\boldsymbol{x} - \boldsymbol{A}s) &= -2(\boldsymbol{x} - \boldsymbol{A}s)^\top \boldsymbol{W}\boldsymbol{A}
\end{aligned}$$

2.9 Error Optimisation

2.9.1 Minimal Optimization Formulation

Given the discussion of basis expansion, we can write a more general version of linear regression (including an expanded basis) as:

$$\hat{y}^{(i)} = \phi(x^{(i)})^T\theta \tag{2.79}$$

The critical learning question in this model is how do we pick the best θ? Unfortunately, outside of linear models, this question does not always have a straight-forward answer and depends on several critical modelling decisions. However, for now, we will take the most basic approach, which is the standard, frequentist learning approach. This is where we model the "best" θ as the one that minimizes a loss or error function, $\mathcal{L}$.

2.9.2 Loss Function

One such loss function is:

$$\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = ||y^{(i)} - \hat{y}^{(i)}||_2^2 \tag{2.80}$$

This is known as the ℓ_2 or mean squared error loss. We can write down learning in the linear regression model as the following optimisation problem:

$$\arg\min_{\theta}[\mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}(y, \hat{y})]] \tag{2.81}$$

That is, the best θ is the value of θ that minimizes the expected loss. Notice that though θ does not appear in the above equation, $\hat{y}$ depends directly on θ.

2.9.3 Learning in a Linear Model

In a linear model, we can express the loss function as:

$$\mathcal{L}(\hat{y}_i, y_i) = (\hat{y}_i - y_i)^2 \tag{2.82}$$

$$\mathcal{L}(\theta) = \frac{1}{N}\left(||\theta^T X - y||_2^2\right) \tag{2.83}$$

Our optimisation problem becomes:

$$\arg\min_{\theta} \mathcal{L}(\theta) \tag{2.84}$$

Note: We can drop the $\frac{1}{N}$ factor because we care about finding the argmin, not the min itself.

2.9.4 Deriving the Optimal Parameter Value

To find the optimal parameter value, we can derive the loss function with respect to θ and set it to zero:

$$\mathcal{L}(\theta) = (\theta^T X - y)^T(\theta^T X - y) \tag{2.85}$$

$$\nabla_\theta(\theta^T X - y)^T(\theta^T X - y) = 0 \tag{2.86}$$

Simplifying:

$$\nabla_\theta||\theta^T X - y||_2^2 = -2X^T(y - \theta^T X) \tag{2.87}$$

$$-2X^T(y - \theta^T X) = 0 \tag{2.88}$$

$$X^T y - X^T X\theta^T = 0 \tag{2.89}$$

$$X^T y = X^T X\theta^T \tag{2.90}$$

Finally, we arrive at the solution:

$$(X^T X)^{-1} X^T y = \theta^T \tag{2.91}$$

This gives us the optimal parameter value for the linear regression model.

What about non-linear models? That's another story...

3
Optimisation and Automatic Differentiation

4
Blah

Theorem 4.0.1 Theorem Name

This is the statement of the theorem.

Corollary 4.0.1 Corollary Name

This is the statement of the corollary.

Lemma 4.0.1 Lemma Name

This is the statement of the lemma.

Claim 4.0.1 Claim Name

This is the statement of the claim.

Example 4.0.1 (Example Name)

This is the explanation of the example.

Note 4.0.1 Side Note Box

This is a side note.

This is a block of highlighted text

Definition Title **Definition 4.0.2**

This is an example definition.

Extra Title ***Non-Examinable* 4.0.2**

This is an example box with extra information.

Example Title **Example Q 4.0.2**

This is an example question.

Answer here

Q1c - 2018 **Exam Q 4.0.2**

This is an example exam question.

Reference Title **Reference 4.0.2**

This is an example reference to source material.

Note 4.0.1 Side Note Box

This is a smaller side note.

This is a block of highlighted text that's smaller.

Small Def Title

Example Text

Definition 4.0.1

Small Title

Example Text

***Non-Examinable* 4.0.1**

Small Title

Example Text

Answer

Example Q 4.0.1

Q1c - 2018

This is an example exam question, smaller.

Exam Q 4.0.1

Reference Title

This is an example reference to source material.

Reference 4.0.1

Intuition Title **Intuition 4.0.2**

This is an example of an intuitive explanation.

Intuition Title

This is an example of an intuitive explanation, smaller

Intuition 4.0.1

THE FRONT MATTER of a book refers to all of the material that comes before the main text. The following table from shows a list of material that appears in the front matter of *The Visual Display of Quantitative Information*, *Envisioning Information*, *Visual Explanations*, and *Beautiful Evidence* along with its page number. Page numbers that appear in parentheses refer to folios that do not have a printed page number (but they are still counted in the page number sequence).

	Books			
Page content	*VDQI*	*EI*	*VE*	*BE*
Blank half title page	(1)	(1)	(1)	(1)
Frontispiece[1]	(2)	(2)	(2)	(2)
Full title page	(3)	(3)	(3)	(3)
Copyright page	(4)	(4)	(4)	(4)
Contents	(5)	(5)	(5)	(5)
Dedication	(6)	(7)	(7)	7
Epigraph	–	–	(8)	–
Introduction	(7)	(9)	(9)	9

[1] The contents of this page vary from book to book. In *VDQI* this page is blank; in *EI* and *VE* this page holds a frontispiece; and in *BE* this page contains three epigraphs.

On the side note of...

yeah

***Non-Examinable* 4.0.3**

The design of the front matter in Tufte's books varies slightly from the traditional design of front matter. First, the pages in front matter are traditionally numbered with lowercase roman numerals (*e. g.*, i, ii, iii, iv, ...). Second, the front matter page numbering sequence is usually separate from the main matter page numbering. That is, the page numbers restart at 1 when the main matter begins. In contrast, Tufte has enumerated his pages with arabic numerals that share the same page counting sequence as the main matter.

There are also some variations in design across Tufte's four books. The page opposite the full title page (labeled "frontispiece" in the above table) has different content in each of the books. In *The Visual Display of Quantitative Information*, this page is blank; in *Envisioning Information* and *Visual Explanations*, this page holds a frontispiece; and in *Beautiful Evidence*, this page contains three epigraphs. The dedication appears on page 6 in *VDQI* (opposite the introduction), and is placed on its own spread in the other books. In *VE*, an epigraph shares the spread with the opening page of the introduction.

None of the page numbers (folios) of the front matter are expressed except in *BE*, where the folios start to appear on the dedication page.

THE FULL TITLE PAGE of each of the books varies slightly in design. In all the books, the author's name appears at the top of the page, the title it set just above the center line, and the publisher is printed along the bottom margin. Some of the differences are outlined in the following table.

Feature	*VDQI*	*EI*	*VE*	*BE*
Author				
Typeface	serif	serif	serif	sans serif
Style	italics	italics	italics	upright, caps
Size	24 pt	20 pt	20 pt	20 pt
Title				
Typeface	serif	serif	serif	sans serif
Style	upright	italics	upright	upright, caps
Size	36 pt	48 pt	48 pt	36 pt
Subtitle				
Typeface	–	–	serif	–
Style	–	–	upright	–
Size	–	–	20 pt	–
Edition				
Typeface	sans serif	–	–	–
Style	upright, caps	–	–	–
Size	14 pt	–	–	–
Publisher				
Typeface	serif	serif	serif	sans serif
Style	italics	italics	italics	upright, caps
Size	14 pt	14 pt	14 pt	14 pt

THE TABLES OF CONTENTS in Tufte's books give us our first glimpse of the structure of the main matter. *The Visual Display of Quantitative Information* is split into two parts, each containing some number of chapters. His other three books only contain chapters—they're not broken into parts.

4.1 Typefaces

Tufte's books primarily use two typefaces: Bembo and Gill Sans. Bembo is used for the headings and body text, while Gill Sans is used for the title page and opening epigraphs in *Beautiful Evidence*.
Since neither Bembo nor Gill Sans are available in default LaTeX installations, the Tufte-LaTeX document classes default to using Palatino and Helvetica, respectively. In addition, the Bera Mono typeface is used for `monospaced` type.
The following font sizes are defined by the Tufte-LaTeX classes:

LaTeX size	Font size	Leading	Used for
`\tiny`	5	6	sidenote numbers
`\scriptsize`	7	8	–
`\footnotesize`	8	10	sidenotes, captions
`\small`	9	12	quote, quotation, and verse environments
`\normalsize`	10	14	body text
`\large`	11	15	B-heads
`\Large`	12	16	A-heads, TOC entries, author, date
`\LARGE`	14	18	handout title
`\huge`	20	30	chapter heads
`\Huge`	24	36	part titles

Table 4.1: A list of LaTeX font sizes as defined by the Tufte-LaTeX document classes.

4.2 Headings

Tufte's books include the following heading levels: parts, chapters,[2] sections, subsections, and paragraphs. Not defined by default are: sub-subsections and subparagraphs.

[2] Parts and chapters are defined for the `tufte-book` class only.

Paragraph Paragraph headings (as shown here) are introduced by italicized text and separated from the main paragraph by a bit of space.

Heading	Style	Size
Part	roman	24/36×40 pc
Chapter	italic	20/30×40 pc
Section	italic	12/16×26 pc
Subsection	italic	11/15×26 pc
Paragraph	italic	10/14

Table 4.2: Heading styles used in *Beautiful Evidence*.

4.3 Environments

The following characteristics define the various environments:

Environment	Font size	Notes
Body text	10/14×26 pc	
Block quote	9/12×24 pc	Block indent (left and right) by 1 pc
Sidenotes	8/10×12 pc	Sidenote number is set inline, followed by word space
Captions	8/10×12 pc	

Table 4.3: Environment styles used in *Beautiful Evidence*.

Column 1	**Column 2**	**Column 3**
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.	Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.	Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.	Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.	Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.	Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.	Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

Table 4.4: Example table with limited column widths

5
On the Use of the `tufte-book` Document Class

The Tufte-LaTeX document classes define a style similar to the style Edward Tufte uses in his books and handouts. Tufte's style is known for its extensive use of sidenotes, tight integration of graphics with text, and well-set typography. This document aims to be at once a demonstration of the features of the Tufte-LaTeX document classes and a style guide to their use.

5.1 Page Layout

5.1.1 Headings

This style provides A- and B-heads (that is, `\section` and `\subsection`), demonstrated above.
If you need more than two levels of section headings, you'll have to define them yourself at the moment; there are no pre-defined styles for anything below a `\subsection`. As Bringhurst points out in *The Elements of Typographic Style*,[1] you should "use as many levels of headings as you need: no more, and no fewer." The Tufte-LaTeX classes will emit an error if you try to use `\subsubsection` and smaller headings.

[1] **Bringhurst2005**

In his later books,[2] Tufte starts each section with a bit of vertical space, a non-indented paragraph, and sets the first few words of the sentence in small caps. To accomplish this using this style, use the `\newthought` command:

[2] **Tufte2006**

```
\newthought{In his later books}, Tufte starts...
```

5.2 Sidenotes

One of the most prominent and distinctive features of this style is the extensive use of sidenotes. There is a wide margin to provide ample room for sidenotes and small figures. Any `\footnote`s will automatically be converted to sidenotes.[3] If you'd like to place ancillary information in the margin without the sidenote mark (the superscript number), you can use the `\marginnote` command.
The specification of the `\sidenote` command is:

[3] This is a sidenote that was entered using the `\footnote` command.

This is a margin note. Notice that there isn't a number preceding the note, and there is no number in the main text where this note was written.

`\sidenote[`⟨*number*⟩`][`⟨*offset*⟩`]{`*Sidenote text.*`}`

Both the ⟨*number*⟩ and ⟨*offset*⟩ arguments are optional. If you provide a ⟨*number*⟩ argument, then that number will be used as the sidenote number. It will change of the number of the current sidenote only and will not affect the numbering sequence of subsequent sidenotes.
Sometimes a sidenote may run over the top of other text or graphics in the margin space. If this happens, you can adjust the vertical position of the sidenote by providing a dimension in the ⟨*offset*⟩ argument. Some examples of valid dimensions are:

```
1.0in     2.54cm     254mm     6\baselineskip
```

If the dimension is positive it will push the sidenote down the page; if the dimension is negative, it will move the sidenote up the page.

While both the ⟨*number*⟩ and ⟨*offset*⟩ arguments are optional, they must be provided in order. To adjust the vertical position of the sidenote while leaving the sidenote number alone, use the following syntax:

`\sidenote[][`⟨*offset*⟩`]{`*Sidenote text.*`}`

The empty brackets tell the `\sidenote` command to use the default sidenote number.
If you *only* want to change the sidenote number, however, you may completely omit the ⟨*offset*⟩ argument:

`\sidenote[`⟨*number*⟩`]{`*Sidenote text.*`}`

The `\marginnote` command has a similar *offset* argument:

`\marginnote[`⟨*offset*⟩`]{`*Margin note text.*`}`

5.3 References

References are placed alongside their citations as sidenotes, as well. This can be accomplished using the normal `\cite` command.[4]
The complete list of references may also be printed automatically by using the `\bibliography` command. (See the end of this document for an example.) If you do not want to print a bibliography at the end of your document, use the `\nobibliography` command in its place.
To enter multiple citations at one location,[5] you can provide a list of keys separated by commas and the same optional vertical offset argument: `\cite{Tufte2006,Tufte1990}`.

`\cite[`⟨*offset*⟩`]{`*bibkey1,bibkey2,...*`}`

[4] The first paragraph of this document includes a citation.

[5] **Tufte2006**, **Tufte1990**

5.4 Figures and Tables

Images and graphics play an integral role in Tufte's work. In addition to the standard `figure` and `tabular` environments, this style provides special figure and table environments for full-width floats.
Full page–width figures and tables may be placed in `figure*` or `table*` environments. To place figures or tables in the margin, use the `marginfigure` or `margintable` environments as follows (see figure 5.1):

```
\begin{marginfigure}
    \includegraphics{helix}
    \caption{This is a margin figure.}
    \label{fig:marginfig}
\end{marginfigure}
```

Figure 5.1: This is a margin figure. The helix is defined by $x = \cos(2\pi z)$, $y = \sin(2\pi z)$, and $z = [0, 2.7]$. The figure was drawn using Asymptote (`http://asymptote.sf.net/`).

The `marginfigure` and `margintable` environments accept an optional parameter ⟨*offset*⟩ that adjusts the vertical position of the figure or table. See the "Sidenotes" section above for examples. The specifications are:

```
\begin{marginfigure}[⟨offset⟩]
    ...
\end{marginfigure}

\begin{margintable}[⟨offset⟩]
    ...
\end{margintable}
```

Figure **??** is an example of the `figure*` environment and figure **??** is an example of the normal `figure` environment.

As with sidenotes and marginnotes, a caption may sometimes require vertical adjustment. The `\caption` command now takes a second optional argument that enables you to do this by providing a dimension ⟨*offset*⟩. You may specify the caption in any one of the following forms:

`\caption`{*long caption*}
`\caption[`*short caption*`]`{*long caption*}
`\caption[]`[⟨*offset*⟩]{*long caption*}
`\caption[`*short caption*`]`[⟨*offset*⟩]{*long caption*}

A positive ⟨*offset*⟩ will push the caption down the page. The short caption, if provided, is what appears in the list of figures/tables, otherwise the "long" caption appears there. Note that although the arguments ⟨*short caption*⟩ and ⟨*offset*⟩ are both optional, they must be provided in order. Thus, to specify an ⟨*offset*⟩ without specifying a ⟨*short caption*⟩, you must include the first set of empty brackets `[]`, which tell `\caption` to use the default "long" caption. As an example, the caption to figure **??** above was given in the form

```
\caption[Hilbert curves...][6pt]{Hilbert curves...}
```

Table 5.1 shows table created with the `booktabs` package. Notice the lack of vertical rules—they serve only to clutter the table's data.

Margin	Length
Paper width	8½ inches
Paper height	11 inches
Textblock width	6½ inches
Textblock/sidenote gutter	⅜ inches
Sidenote width	2 inches

Table 5.1: Here are the dimensions of the various margins used in the Tufte-handout class.

Occasionally LaTeX will generate an error message:

```
Error:  Too many unprocessed floats
```

LaTeX tries to place floats in the best position on the page. Until it's finished composing the page, however, it won't know where those positions are. If you have a lot of floats on a page (including sidenotes, margin notes, figures, tables, etc.), LaTeX may run out of "slots" to keep track of them and will generate the above error.

LaTeX initially allocates 18 slots for storing floats. To work around this limitation, the Tufte-LaTeX document classes provide a `\morefloats` command that will reserve more slots.

The first time `\morefloats` is called, it allocates an additional 34 slots. The second time `\morefloats` is called, it allocates another 26 slots.

The `\morefloats` command may only be used two times. Calling it a third time will generate an error message. (This is because we can't safely allocate many more floats or LaTeX will run out of memory.)

If, after using the `\morefloats` command twice, you continue to get the `Too many unprocessed floats` error, there are a couple things you can do.

The `\FloatBarrier` command will immediately process all the floats before typesetting more material. Since `\FloatBarrier` will start a new paragraph, you should place this command at the beginning or end of a paragraph.

The `\clearpage` command will also process the floats before continuing, but instead of starting a new paragraph, it will start a new page.

You can also try moving your floats around a bit: move a figure or table to the next page or reduce the number of sidenotes. (Each sidenote actually uses *two* slots.)

After the floats have placed, LaTeX will mark those slots as unused so they are available for the next page to be composed.

5.5 Captions

You may notice that the captions are sometimes misaligned. Due to the way LaTeX's float mechanism works, we can't know for sure where it decided to put a float. Therefore, the Tufte-LaTeX document classes provide commands to override the caption position.

Vertical alignment To override the vertical alignment, use the `\setfloatalignment` command inside the float environment. For example:

```
\begin{figure}[btp]
    \includegraphics{sinewave}
    \caption{This is an example of a sine wave.}
    \label{fig:sinewave}
    \setfloatalignment{b}% forces caption to be bottom-aligned
\end{figure}
```

The syntax of the `\setfloatalignment` command is:

```
\setfloatalignment{⟨pos⟩}
```

where ⟨*pos*⟩ can be either `b` for bottom-aligned captions, or `t` for top-aligned captions.

Horizontal alignment To override the horizontal alignment, use either the `\forceversofloat` or the `\forcerectofloat` command inside of the float environment. For example:

```
\begin{figure}[btp]
    \includegraphics{sinewave}
    \caption{This is an example of a sine wave.}
    \label{fig:sinewave}
    \forceversofloat% forces caption to be set to the left of the float
\end{figure}
```

The `\forceversofloat` command causes the algorithm to assume the float has been placed on a verso page—that is, a page on the left side of a two-page spread. Conversely, the `\forcerectofloat` command causes the algorithm to assume the float has been placed on a recto page—that is, a page on the right side of a two-page spread.

5.6 Full-width text blocks

In addition to the new float types, there is a `fullwidth` environment that stretches across the main text block and the sidenotes area.

```
\begin{fullwidth}
  Lorem ipsum dolor sit amet...
\end{fullwidth}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

5.7 Typography

5.7.1 Typefaces

If the Palatino, Helvetica, and `Bera Mono` typefaces are installed, this style will use them automatically. Otherwise, we'll fall back on the Computer Modern typefaces.

5.7.2 Letterspacing

This document class includes two new commands and some improvements on existing commands for letterspacing.

When setting strings of ALL CAPS or SMALL CAPS, the letterspacing—that is, the spacing between the letters—should be increased slightly.[6] The `\allcaps` command has proper letterspacing for strings of FULL CAPITAL LETTERS, and the `\smallcaps` command has letterspacing for SMALL CAPITAL LETTERS. These commands will also automatically convert the case of the text to upper- or lowercase, respectively.

[6] **Bringhurst2005**

The `\textsc` command has also been redefined to include letterspacing. The case of the `\textsc` argument is left as is, however. This allows one to use both uppercase and lowercase letters: The Initial Letters Of The Words In This Sentence Are Capitalized.

5.8 Document Class Options

The `tufte-book` class is based on the LaTeX `book` document class. Therefore, you can pass any of the typical book options. There are a few options that are specific to the `tufte-book` document class, however.

The `a4paper` option will set the paper size to A4 instead of the default US letter size.

The `sfsidenotes` option will set the sidenotes and title block in a sans serif typeface instead of the default roman.

The `twoside` option will modify the running heads so that the page number is printed on the outside edge (as opposed to always printing the page number on the right-side edge in `oneside` mode).

The `symmetric` option typesets the sidenotes on the outside edge of the page. This is how books are traditionally printed, but is contrary to Tufte's book design which sets the sidenotes on the right side of the page. This option implicitly sets the `twoside` option.

The `justified` option sets alldocclsoptdef and right). The default is to set the text ragged right. The body text of Tufte's books are set ragged right. This prevents needless hyphenation and makes it easier to read the text in the slightly narrower column.

The `bidi` option loads the `bidi` package which is used with XeLaTeX to typeset bi-directional text. Since the `bidi` package needs to be loaded before the sidenotes and cite commands are defined, it can't be loaded in the document preamble.

The `debug` option causes the Tufte-LaTeX classes to output debug information to the log file which is useful in troubleshooting bugs. It will also cause the graphics to be replaced by outlines.

The `nofonts` option prevents the Tufte-LaTeX classes from automatically loading the Palatino and Helvetica typefaces. You should use this option if you wish to load your own fonts. If you're using XeLaTeX, this option is implied (*i. e.*, the Palatino and Helvetica fonts aren't loaded if you use XeLaTeX).

The `nols` option inhibits the letterspacing code. The Tufte-LaTeX classes try to load the appropriate letterspacing package (either pdfTeX's `letterspace` package

or the `soul` package). If you're using XeLaTeX with `fontenc`, however, you should configure your own letterspacing.

The `notitlepage` option causes `\maketitle` to generate a title block instead of a title page. The `book` class defaults to a title page and the `handout` class defaults to the title block. There is an analogous `titlepage` option that forces `\maketitle` to generate a full title page instead of the title block.

The `notoc` option suppresses Tufte-LaTeX's custom table of contents (TOC) design. The current TOC design only shows unnumbered chapter titles; it doesn't show sections or subsections. The `notoc` option will revert to LaTeX's TOC design.

The `nohyper` option prevents the `hyperref` package from being loaded. The default is to load the `hyperref` package and use the `\title` and `\author` contents as metadata for the generated PDF.

6
Customizing Tufte-LaTeX

The Tufte-LaTeX document classes are designed to closely emulate Tufte's book design by default. However, each document is different and you may encounter situations where the default settings are insufficient. This chapter explores many of the ways you can adjust the Tufte-LaTeX document classes to better fit your needs.

6.1 File Hooks

If you create many documents using the Tufte-LaTeX classes, it's easier to store your customizations in a separate file instead of copying them into the preamble of each document. The Tufte-LaTeX classes provide three file hooks: `tufte-common-local.tex`, `tufte-book-local.tex`, and `tufte-handout-local.tex`.

`tufte-common-local.tex` If this file exists, it will be loaded by all of the Tufte-LaTeX document classes just prior to any document-class-specific code. If your customizations or code should be included in both the book and handout classes, use this file hook.

`tufte-book-local.tex` If this file exists, it will be loaded after all of the common and book-specific code has been read. If your customizations apply only to the book class, use this file hook.

`tufte-common-handout.tex` If this file exists, it will be loaded after all of the common and handout-specific code has been read. If your customizations apply only to the handout class, use this file hook.

6.2 Numbered Section Headings

While Tufte dispenses with numbered headings in his books, if you require them, they can be anabled by changing the value of the `secnumdepth` counter. From the table below, select the heading level at which numbering should stop and set the `secnumdepth` counter to that value. For example, if you want parts and chapters numbered, but don't want numbering for sections or subsections, use the command:

```
\setcounter{secnumdepth}{0}
```

The default `secnumdepth` for the Tufte-LaTeX document classes is −1.

Heading level	Value
Part (in `tufte-book`)	−1
Part (in `tufte-handout`)	0
Chapter (only in `tufte-book`)	0
Section	1
Subsection	2
Subsubsection	3
Paragraph	4
Subparagraph	5

Table 6.1: Heading levels used with the `secnumdepth` counter.

6.3 Changing the Paper Size

The Tufte-LaTeX classes currently only provide three paper sizes: A4, B5, and US letter. To specify a different paper size (and/or margins), use the `\geometrysetup` command in the preamble of your document (or one of the file hooks). The full documentation of the `\geometrysetup` command may be found in the `geometry` package documentation.[1]

[1] **pkg-geometry**

6.4 Customizing Marginal Material

Marginal material includes sidenotes, citations, margin notes, and captions. Normally, the justification of the marginal material follows the justification of the body text. If you specify the `justified` document class option, all of the margin material will be fully justified as well. If you don't specify the `justified` option, then the marginal material will be set ragged right.

You can set the justification of the marginal material separately from the body text using the following document class options: `sidenote`, `marginnote`, `caption`, `citation`, and `marginals`. Each option refers to its obviously corresponding marginal material type. The `marginals` option simultaneously sets the justification on all four marginal material types.

Each of the document class options takes one of five justification types:

`justified` Fully justifies the text (sets it flush left and right).

`raggedleft` Sets the text ragged left, regardless of which page it falls on.

`raggedright` Sets the text ragged right, regardless of which page it falls on.

`raggedouter` Sets the text ragged left if it falls on the left-hand (verso) page of the spread and otherwise sets it ragged right. This is useful in conjunction with the `symmetric` document class option.

`auto` If the `justified` document class option was specified, then set the text fully justified; otherwise the text is set ragged right. This is the default justification option if one is not explicitly specified.

For example,

```
\documentclass[symmetric,justified,marginals=raggedouter]{tufte-book}
```

will set the body text of the document to be fully justified and all of the margin material (sidenotes, margin notes, captions, and citations) to be flush against the body text with ragged outer edges.

THE FONT AND STYLE of the marginal material may also be modified using the following commands:

```
\setsidenotefont{⟨font commands⟩}
\setcaptionfont{⟨font commands⟩}
\setmarginnotefont{⟨font commands⟩}
\setcitationfont{⟨font commands⟩}
```

The `\setsidenotefont` sets the font and style for sidenotes, the `\setcaptionfont` for captions, the `\setmarginnotefont` for margin notes, and the `\setcitationfont` for citations. The ⟨*font commands*⟩ can contain font size changes (e.g., `\footnotesize`, `\Huge`, etc.), font style changes (e.g., `\sffamily`, `\ttfamily`, `\itshape`, etc.), color changes (e.g., `\color{blue}`), and many other adjustments.

If, for example, you wanted the captions to be set in italic sans serif, you could use:

```
\setcaptionfont{\itshape\sffamily}
```

7
Compatibility Issues

When switching an existing document from one document class to a Tufte-LaTeX document class, a few changes to the document may have to be made.

7.1 Converting from `article` to `tufte-handout`

The following `article` class options are unsupported: `10pt`, `11pt`, `12pt`, `a5paper`, `b5paper`, `executivepaper`, `legalpaper`, `landscape`, `onecolumn`, and `twocolumn`. The following headings are not supported: `\subsubsection` and `\subparagraph`.

7.2 Converting from `book` to `tufte-book`

The following `report` class options are unsupported: `10pt`, `11pt`, `12pt`, `a5paper`, `b5paper`, `executivepaper`, `legalpaper`, `landscape`, `onecolumn`, and `twocolumn`. The following headings are not supported: `\subsubsection` and `\subparagraph`.

8
Troubleshooting and Support

8.1 Tufte-LaTeX Website

The website for the Tufte-LaTeX packages is located at http://code.google.com/p/tufte-latex/. On our website, you'll find links to our SVN repository, mailing lists, bug tracker, and documentation.

8.2 Tufte-LaTeX Mailing Lists

There are two mailing lists for the Tufte-LaTeX project:

Discussion list The `tufte-latex` discussion list is for asking questions, getting assistance with problems, and help with troubleshooting. Release announcements are also posted to this list. You can subscribe to the `tufte-latex` discussion list at http://groups.google.com/group/tufte-latex.

Commits list The `tufte-latex-commits` list is a read-only mailing list. A message is sent to the list any time the Tufte-LaTeX code has been updated. If you'd like to keep up with the latest code developments, you may subscribe to this list. You can subscribe to the `tufte-latex-commits` mailing list at http://groups.google.com/group/tufte-latex-commits.

8.3 Getting Help

If you've encountered a problem with one of the Tufte-LaTeX document classes, have a question, or would like to report a bug, please send an email to our mailing list or visit our website.
To help us troubleshoot the problem more quickly, please try to compile your document using the `debug` class option and send the generated `.log` file to the mailing list with a brief description of the problem.

8.4 Errors, Warnings, and Informational Messages

The following is a list of all of the errors, warnings, and other messages generated by the Tufte-LaTeX classes and a brief description of their meanings.

`Error: \subparagraph is undefined by this class.`

The `\subparagraph` command is not defined in the Tufte-LaTeX document classes. If you'd like to use the `\subparagraph` command, you'll need to redefine it yourself. See the "Headings" section on page 37 for a description of the heading styles availaboe in the Tufte-LaTeX document classes.

`Error: \subsubsection is undefined by this class.`

The `\subsubsection` command is not defined in the Tufte-LaTeX document classes. If you'd like to use the `\subsubsection` command, you'll need to redefine

it yourself. See the "Headings" section on page 37 for a description of the heading styles availaboe in the Tufte-LaTeX document classes.

```
Error: You may only call \morefloats twice. See the
       Tufte-LaTeX documentation for other workarounds.
```

LaTeX allocates 18 slots for storing floats. The first time `\morefloats` is called, it allocates an additional 34 slots. The second time `\morefloats` is called, it allocates another 26 slots.
The `\morefloats` command may only be called two times. Calling it a third time will generate this error message. See page 39 for more information.

```
Warning: Option '⟨class option⟩' is not supported -- ignoring option.
```

This warning appears when you've tried to use ⟨*class option*⟩ with a Tufte-LaTeX document class, but ⟨*class option*⟩ isn't supported by the Tufte-LaTeX document class. In this situation, ⟨*class option*⟩ is ignored.

```
Info: The 'symmetric' option implies 'twoside'
```

You specified the `symmetric` document class option. This option automatically forces the `twoside` option as well. See page 41 for more information on the `symmetric` class option.

8.5 Package Dependencies

The following is a list of packages that the Tufte-LaTeX document classes rely upon. Packages marked with an asterisk are optional.

- xifthen
- ifpdf*
- ifxetex*
- hyperref
- geometry
- ragged2e
- chngpage *or* changepage
- paralist
- textcase
- soul*
- letterspace*
- setspace
- natbib *and* bibentry
- optparams
- placeins
- mathpazo*
- helvet*
- fontenc
- beramono*
- fancyhdr
- xcolor
- textcomp
- titlesec
- titletoc