



CLUSTERING

Theoretical and Practical Aspects

Unlike the union, the intersection is defined only for collections that consist of subsets of a set S .

If \mathcal{C} is a collection of subsets of S , that is, if $\mathcal{C} \subseteq \mathcal{P}(S)$ then the intersection of \mathcal{C} is the set of all elements of S that belong to every set of \mathcal{C} . The intersection of \mathcal{C} is denoted by $\bigcap \mathcal{C}$.

If \mathcal{C} and \mathcal{C}' are two collections of subsets of a set S and $\mathcal{C} \subseteq \mathcal{C}'$, then $\bigcap \mathcal{C}' \subseteq \bigcap \mathcal{C}$.

If \emptyset is the empty collection of subsets of S , we define $\bigcap \emptyset = S$.

Definition 2.7. A *closure system on the set S* is a collection \mathcal{K} of subsets of S such that for every collection of subsets \mathcal{C} such that $\mathcal{C} \subseteq \mathcal{K}$ we have $\bigcap \mathcal{C} \in \mathcal{K}$.

Note that if \mathcal{K} is a closure system on a set S , then $S \in \mathcal{K}$ because S is the intersection of the empty collection of subsets of \mathcal{K} .

Definition 2.8. Let \mathcal{K} be a closure system on a set S and let T be a subset of S . The *closure of T relative to the closure system \mathcal{K}* is the set $\mathbf{K}(T) = \bigcap \{U \in \mathcal{K} \mid T \subseteq U\}$.

For every set T the collection $\mathcal{C}_T = \{U \in \mathcal{K} \mid T \subseteq U\}$ is non-empty because it includes at least S . The set $\bigcap \mathcal{C}_T$ is denoted by $\mathbf{K}(T)$ and is referred to as the *closure* of T .

To emphasize that the closure of T is computed relative to the closure system \mathcal{K} we may denote this closure by $\mathbf{K}_{\mathcal{K}}(T)$.

Example 2.7. A subset E of \mathbb{R} is said to be symmetric if $x \in E$ if and only if $-x \in E$.

Let $\{E_i \mid i \in I\}$ be a collection of symmetric subsets of \mathbb{R} . It is easy to see that $\bigcap \{E_i \mid i \in I\}$ is a symmetric set. Note that \mathbb{R} itself is symmetric. Thus, the collection \mathcal{E} of symmetric subsets of \mathbb{R} is a closure system. For a subset T of \mathbb{R} the set $\mathbf{K}_{\mathcal{E}}(T)$ is the smallest symmetric set that includes T .

The notion of *closure operator* can be defined independently.

Definition 2.9. A *closure operator* on set S is a mapping $\mathbf{K} : \mathcal{P}(S) \longrightarrow \mathcal{P}(S)$ that has the following properties:

- (i) $X \subseteq \mathbf{K}(X)$ (extensivity);
- (ii) $\mathbf{K}(\mathbf{K}(X)) = \mathbf{K}(X)$ (idempotency);
- (iii) $X \subseteq X'$ implies $\mathbf{K}(X) \subseteq \mathbf{K}(X')$ (monotonicity).

This page intentionally left blank

CLUSTERING

Theoretical and Practical Aspects

Dan A Simovici

University of Massachusetts Boston, USA

 **World Scientific**

NEW JERSEY • LONDON • SINGAPORE • BEIJING • SHANGHAI • HONG KONG • TAIPEI • CHENNAI • TOKYO

The *intersection* of M and N is the multiset $M \cap N$ defined by

$$(M \cap N)(x) = \min\{M(x), N(x)\}$$

for $x \in S$.

The *sum* of M and N is the multiset $M + N$ given by

$$(M + N)(x) = M(x) + N(x)$$

for $x \in S$.

Let S be a multiset and let U be a sub-multiset of S . The *complement* of U relative to S is the sub-multiset $V = S - U$ of S defined by $m_V(x) = m_S(x) - m_U(x)$ for $x \in S$.

Example 2.6. Let $m, n \in \mathbb{N}$ be two numbers that have the prime factorizations

$$m = p_{i_1}^{k_1} \cdots p_{i_r}^{k_r},$$

$$n = p_{j_1}^{h_1} \cdots p_{j_s}^{h_s},$$

and let M_m, M_n be the multisets of their prime divisors, as defined in Example 2.5. Denote by $\gcd(m, n)$ the greatest common divisor of m and n , and by $\text{lcm}(m, n)$ the least common multiple of these numbers.

We have

$$M_{\gcd(m, n)} = M_m \cap M_n,$$

$$M_{\text{lcm}(m, n)} = M_m \cup M_n,$$

$$M_{mn} = M_m + M_n,$$

as the reader can easily verify.

Definition 2.6. Let M be a multiset. Its *cardinality* is the number $|\{x \in S \mid M(x) \geq 0\}|$; its *size* is $|M| = \sum \{M(x) \mid x \in S\}$.

A multiset on the set $\mathcal{P}(S)$ is referred to as a *multicollection* of sets on S .

2.5 Closure Systems

Let $\mathcal{C} = \{S_i \mid i \in I\}$ be a collection of sets. Its union is the set U defined as

$$U = \bigcup_{i \in I} S_i.$$

Note that $\mathcal{C} \subseteq \mathcal{C}'$ implies $\bigcup \mathcal{C} \subseteq \bigcup \mathcal{C}'$.

Example 2.5. Let PRIMES be the set of prime numbers:

$$\text{PRIMES} = \{2, 3, 5, 7, 11, \dots\}. \quad (2.3)$$

A number is determined by the multiset of its prime divisors in the

following sense. If $n \in \mathbb{N}$, $n \geq 1$, can be factored as a product of prime numbers, $n = p_{i_1}^{k_1} \cdots p_{i_\ell}^{k_\ell}$, where p_i is the i^{th} prime number and k_1, \dots, k_ℓ are positive numbers, then the multiset of its prime divisors is the multiset $M_n : \text{PRIMES} \rightarrow \mathbb{N}$, where $M_n(p)$ is the exponent of the prime number p in the product (2.3).

For example, M_{1960} is given by

$$M_{1960}(p) = \begin{cases} 3 & \text{if } p = 2, \\ 1 & \text{if } p = 5, \\ 2 & \text{if } p = 7. \end{cases}$$

Thus, $\text{car}_r(M_{1960}) = \{2, 5, 7\}$.

Note that if $m, n \in \mathbb{N}$, we have $M_m = M_n$ if and only if $m = n$.

We denote a multiset by using square brackets instead of braces. If x has the multiplicity n in a multiset M , we write x a number of times n inside the square brackets. For example, the multiset of Example 2.5 can be written as $[2, 2, 2, 5, 7, 7]$.

Note that while multiplicity counts in a multiset, order does not matter; therefore, the multiset $[2, 2, 2, 5, 7, 7]$ could also be denoted by $[5, 2, 7, 2, 2, 2]$ or $[7, 5, 2, 7, 2, 2]$. We also use the abbreviation $n * x$ in a multiset to mean that x has the multiplicity n in M . For example, the multiset M_{1960} can be written as $M_{1960} = [3 * 2, 1 * 5, 2 * 7]$.

The multiset M on the set S defined by $M(x) = 0$ for $x \in S$ is the *empty multiset*.

Let U and V be two multisets on a set S . U is a sub-multiset of V if $U(x) \leq V(x)$ for every $x \in S$.

Multisets can be combined to construct new multisets. Common set-theoretical operations such as union and intersection have natural generalizations to multisets.

Definition 2.5. Let M and N be two multisets on a set S .

The *union* of M and N is the multiset $M \cup N$ defined by

$$(M \cup N)(x) = \max\{M(x), N(x)\}$$

for $x \in S$.

CLUSTERING

Theoretical and Practical Aspects

Copyright © 2022 by World Scientific Publishing Co. Pte. Ltd.

All rights reserved. This book, or parts thereof, may not be reproduced in any form or by any means, electronic or mechanical, including photocopying, recording or any information storage and retrieval system now known or to be invented, without written permission from the publisher.

For photocopying of material in this volume, please pay a copying fee through the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, USA. In this case permission to photocopy is not required from the publisher.

ISBN 978-981-124-119-2 (hardcover)

ISBN 978-981-124-120-8 (ebook for institutions)

ISBN 978-981-124-121-5 (ebook for individuals)

For any available supplementary material, please visit

<https://www.worldscientific.com/worldscibooks/10.1142/12394#t=suppl>

The sequence \mathbf{s}_n is the *state* of S at moment n .

Example 2.4. Let $A = \{a, b, c\}$, $\mathbf{q}_0 = \lambda$. By pushing a, b, b, c starting from $\mathbf{s}_0 = \lambda$ we obtain

$$\begin{aligned}\mathbf{s}_1 &= \text{push}(\mathbf{s}_0, a) = (a), \\ \mathbf{s}_2 &= \text{push}(\mathbf{s}_1, b) = (b, a), \\ \mathbf{s}_3 &= \text{push}(\mathbf{s}_2, b) = (b, b, a), \\ \mathbf{s}_4 &= \text{push}(\mathbf{s}_3, c) = (c, b, b, a).\end{aligned}$$

When the **pop** operation is applied, elements are extracted from the left end of the sequence. This yields:

$$\begin{aligned}\text{pop}(\mathbf{s}_4) &= (\mathbf{s}_5, c), \mathbf{s}_5 = (b, b, a) \\ \text{pop}(\mathbf{s}_5) &= (\mathbf{s}_6, b), \mathbf{s}_6 = (b, a) \\ \text{pop}(\mathbf{s}_6) &= (\mathbf{s}_7, b), \mathbf{s}_7 = (a) \\ \text{pop}(\mathbf{s}_7) &= (\mathbf{s}_8, a), \mathbf{s}_8 = \lambda.\end{aligned}$$

Note that the elements of the stack are extracted in the reverse order of their push on the stack, c, b, b, a .

Thus, the working of a stack can be described by “first-in last-out” rule.

2.4 Multisets

Multisets generalize the notion of a set by allowing multiple copies of an element. Formally, we have the following definition.

Definition 2.4. A *multiset* on a set S is a function $M : S \rightarrow \mathbb{N}$. Its *carrier* is the set $\text{carr}(M) = \{x \in S \mid M(x) > 0\}$.

The *multiplicity* of an element x of S in the multiset M is the number $M(x)$.

The set of all multisets on S is denoted by $\mathcal{M}(S)$.

Note that a subset T of S can be regarded as a multiset $T : S \rightarrow \mathbb{N}$, where

$$T(x) = \begin{cases} 1 & \text{if } x \in T, \\ 0 & \text{otherwise,} \end{cases}$$

for $x \in S$.

To my wife Doina,
and to the memory of my parents,
Adelina and Avram Simovici

Sequences allow us to define two algorithmic concepts, namely, queues and stacks.

Definition 2.2. Let A be a set. A *queue* on A is a triple $\mathbf{q} = (Q, e, d)$, where Q is a sequence of sequences of A ,

$$Q = (\mathbf{q}_0, \mathbf{q}_1, \dots),$$

$e : A \times Q \rightarrow Q$ is the enqueueing operation and $d : Q \rightarrow Q \times A$ is the

dequeueing operation are two functions that satisfy the following conditions:

(i) $e(a, \mathbf{q}) = (a, \mathbf{q})$

(ii) $d(\mathbf{q})$ is defined on non-null sequence and $d(\mathbf{q}) = (\mathbf{q}', a)$ if $\mathbf{q} = \mathbf{q}'(a)$.

Example 2.3. Let $A = \{a, b, c\}$, $\mathbf{q}_0 = \lambda$. By enqueueing the a, b, b, c starting from $\mathbf{q}_0 = \lambda$ we obtain

$$\mathbf{q}_1 = e(\mathbf{q}_0, a) = (a),$$

$$\mathbf{q}_2 = e(\mathbf{q}_1, b) = (b, a),$$

$$\mathbf{q}_3 = e(\mathbf{q}_2, b) = (b, b, a),$$

$$\mathbf{q}_4 = e(\mathbf{q}_3, c) = (c, b, b, a).$$

When the dequeueing operation is applied, elements are extracted from the right end of the sequence. This yields the sequence

$$d(\mathbf{q}_4) = (\mathbf{q}_5, a) = (c, b, b)$$

$$d(\mathbf{q}_5) = (\mathbf{q}_6, b) = (c, b)$$

$$d(\mathbf{q}_6) = (\mathbf{q}_7, b) = (c)$$

$$d(\mathbf{q}_7) = (\mathbf{q}_8, c) = \lambda.$$

Note that the working of a queue can be described by the syntagm “first-in first-out”. Indeed, the order in which elements are produced by the dequeueing operation is the same as the order these elements were enqueued: a, b, b, c .

Definition 2.3. Let A be a set. A *stack* on A is a triple $\mathbf{s} = (S, e, d)$, where S is a sequence of sequences of A ,

$$S = (\mathbf{s}_0, \mathbf{s}_1, \dots),$$

$\text{push} : A \times S \rightarrow S$ is the push operation and $\text{pop} : S \rightarrow S \times A$ is the popping operation are two functions that satisfy the following conditions:

(i) $\text{push}(a, \mathbf{s}) = (\mathbf{s}, a)$,

(ii) $\text{pop}(\mathbf{s})$ is defined on non-null sequence and $\text{pop}(\mathbf{s}) = (\mathbf{s}', a)$ if $\mathbf{s} = \mathbf{s}'(a)$.

This page intentionally left blank

Preface

Clustering is a part of machine learning that seeks to identify groups into sets of objects such that objects that belong to the same group are as similar as possible, and objects that belong to two distinct groups are as dissimilar as possible. In general, clustering exploration is based on computing similarities (or dissimilarities) between objects but does not provide the reasons for the existence of these groupings.

Various notions of dissimilarities are considered among objects ranging from simple dissimilarities, metrics on linear spaces, ultrametrics, and extensions of these measures to sets. Studying these measures requires incursions in a variety of mathematical disciplines ranging from linear algebra and optimization to functional analysis and topology.

The results of clusterings are evaluated using a variety of criteria allowing users to choose clusterings that are desirable from the point of view of these criteria.

Clustering use is widespread, ranging from genomics, epidemiology, medicine, economics and many other disciplines. The intended readership of this volume consists of researchers and graduate students who work in data mining and pattern recognition, or apply those in their domain of interest. I strived to make this volume as self-contained as possible. Appendices, exercises, and supplements are provided to help readers in their search of mathematical tools useful for clustering.

Boston and Brookline

Dan A. Simovici
May 2021

2.3 Sequences

Definition 2.1. Let S be a set. A *sequence of length n on S* is a mapping $\mathbf{s} : \{1, \dots, n\} \rightarrow S$. The set of sequences of length n on S is denoted by $\mathbf{Seq}_n(S)$.

An *ordered pair* on S is a sequence of length 2 on S ; a *singleton* is a sequence of length 1.

If \mathbf{s} is a sequence of length n on S and $\mathbf{s}(i) = x_i$ for $1 \leq i \leq n$, we write $\mathbf{s} = (x_1, \dots, x_n)$. The elements x_1, \dots, x_n are the *components* of \mathbf{s} .

The length of a sequence \mathbf{s} is denoted by $|\mathbf{s}|$.

Example 2.1. A sequence of natural numbers of length 6 is $\mathbf{s} = (6, 5, 2, 4, 9, 6)$. Note that in a sequence the same element of S may occur on multiple positions.

If S is a finite set containing m elements, then there are m^n sequences of length n for any $n \geq 1$. We extend the definition of sequences on S by defining the *null sequence on S* as the sequence λ that has no components, $\lambda = ()$. Note that there exists exactly one such sequence on S and this is consistent with the fact that $m^0 = 1$ for every $m \geq 1$.

The *set of sequences of elements of S* is the set

$$\mathbf{Seq}(S) = \bigcup \{\mathbf{Seq}_n(S) \mid n \geq 0\}.$$

If $\mathbf{s} = (s_1, s_2, \dots, s_n)$ is a sequence in S , we refer to the sequence $\tilde{\mathbf{s}} = (s_n, \dots, s_2, s_1)$ as the *reversal* of the sequence \mathbf{s} . Clearly $\tilde{\tilde{\mathbf{s}}} = \mathbf{s}$.

If $\mathbf{s} = (s_1, \dots, s_n)$ and $\mathbf{t} = (t_1, \dots, t_m)$ are two sequences on a set S , their *concatenation* is the sequence $\mathbf{st} = (s_1, \dots, s_n, t_1, \dots, t_m)$. For the null sequence we define $\lambda\mathbf{s} = \mathbf{s}\lambda = \mathbf{s}$ for every $\mathbf{s} \in \mathbf{Seq}(S)$. Note that $|\mathbf{st}| = |\mathbf{s}| + |\mathbf{t}|$ for all sequences $\mathbf{s}, \mathbf{t} \in \mathbf{Seq}(S)$.

Note that sequence concatenation is not a commutative operation in general.

Example 2.2. Let $\mathbf{s} = (1, 2, 3)$, $\mathbf{t} = (4, 5)$. We have

$$\mathbf{st} = (1, 2, 3, 4, 5) \text{ and } \mathbf{ts} = (4, 5, 1, 2, 3),$$

so $\mathbf{st} \neq \mathbf{ts}$.

We leave to the reader to verify that sequence concatenation is an associative operation on $\mathbf{Seq}(S)$, that is $(\mathbf{st})\mathbf{u} = \mathbf{s}(\mathbf{tu})$ for every $\mathbf{s}, \mathbf{t}, \mathbf{u} \in \mathbf{Seq}(S)$.

\oplus	0	1
0	0	1
1	1	0

\cdot	0	1
0	0	0
1	0	1

Let S be a set and let $\mathbf{GF}(2)$ be the two element field defined in earlier. Define the scalar multiplication of a subset T of S by an element of the field as

$$0 \cdot T = \emptyset \text{ and } 1 \cdot T = T$$

for every $T \in \mathcal{P}(S)$.

The sum of two subsets U and V is defined as their symmetric difference

$$U + V = (U - V) \cup (V - U).$$

With these definitions the set $\mathcal{P}(S)$ of subsets of S is an $\mathbf{GF}(2)$ -linear space,

as the reader can easily verify.

The set of subsets $\mathcal{P}(S)$ of a finite set $S = \{x_1, \dots, x_n\}$ can be organized

as a $\mathbf{GF}(2)$ -linear space by defining the sum of two subsets U, V as their

symmetric difference

$$U \oplus V = (U - V) \cup (V - U).$$

Note that $U \oplus \emptyset = \emptyset \oplus U = U$.

Multiplication with scalars in $\{0, 1\}$ is defined as

$$0 \cdot U = \emptyset \text{ and } 1 \cdot U = U,$$

for every $U \in \mathcal{P}(S)$.

A basis in the $\mathbf{GF}(2)$ -linear space of subsets of the set $S = \{x_1, \dots, x_n\}$

is the collection $\{\{x_1\}, \dots, \{x_n\}\}$. Every subset U of S can be uniquely

written as

$$U = a_1\{x_1\} \oplus \dots \oplus a_n\{x_n\},$$

where

$$a_i = \begin{cases} 1 & \text{if } x_i \in U, \\ 0 & \text{if } x_i \notin U, \end{cases}$$

for $1 \leq i \leq n$. Thus, the $\mathbf{GF}(2)$ -linear space of subsets of S is of dimension

n .

The inner product of two vectors $\mathbf{u}, \mathbf{v} \in \mathbf{GF}(2)^n$ is

$$(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{v} = \bigoplus_{i=1}^n u_i v_i.$$

Note that if $\mathbf{u} \in \mathbf{GF}(2)^n$, where n is an even number, we have $(\mathbf{u}, \mathbf{u}) = 0$,

even if $\mathbf{u} \neq \mathbf{0}_n$.

This page intentionally left blank

Contents

<i>Preface</i>	vii
1. Introduction	1
2. Set-Theoretical Preliminaries	5
2.1 Introduction	5
2.2 Sets and Set Operations	5
2.3 Sequences	8
2.4 Multisets	10
2.5 Closure Systems	12
2.6 Permutations	14
2.7 Relations	19
2.8 Partially Ordered Sets	26
2.9 The Poset of Partitions of a Set	38
2.10 Boolean Matrices	43
2.11 Galois Connections and Formal Concepts	56
2.12 Exercises and Supplements	65
2.13 Bibliographical Comments	71
3. Dissimilarities, Metrics, and Ultrametrics	73
3.1 Introduction	73
3.2 Dissimilarity Spaces	73
3.3 Similarities and Dissimilarities between Sets	80
3.4 Norms and Metrics on \mathbb{R}^n	84
3.5 The Geometry of Ultrametric Spaces	96
3.6 Matrices on Semirings	103

have

$$A \oplus B = (A - B) \cup (B - A).$$

For set inclusion we write $A \subseteq B$ to denote that each element x of A also belongs to B .

Note that $A = B$ if and only if $A \oplus B = \emptyset$.

For a set S we denote by $\mathcal{P}(S)$ the set of its subsets. The collection of subsets of S that contain k elements is denoted by $\mathcal{P}_k(S)$. The sets in $\mathcal{P}_2(S)$ are the *unordered pairs* of S .

A subset A of a set S is completely described by its *characteristic function* $\mathbf{1}_A : S \rightarrow \{0, 1\}$ defined as

$$\mathbf{1}_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{otherwise} \end{cases}$$

for $x \in S$.

If $A, B, C \in \mathcal{P}(S)$ we have:

$$A \oplus B = (A \oplus C) \oplus (C \oplus B). \quad (2.1)$$

This equality can be verified by considering all cases determined by the membership of an element x in A, B and C . These are summarized by the following table, where 1 indicates that x belongs to the set and 0 means that x is not a member of the set that labels the each column.

A	B	C	$A \oplus B$	$A \oplus C$	$C \oplus B$	$(A \oplus C) \oplus (C \oplus B)$
0	0	0	0	0	0	0
0	0	1	0	1	1	0
0	1	0	1	0	1	1
0	1	1	1	1	0	1
1	0	0	1	1	0	1
1	0	1	1	0	1	1
1	1	0	0	1	1	0
1	1	1	0	0	0	0

In all cases, the entries of the column $A \oplus B$ coincide with the entries of the column $(A \oplus C) \oplus (C \oplus B)$ proving Equality (2.1). This equality implies immediately

$$|A \oplus B| \leq |A \oplus C| + |C \oplus B| \quad (2.2)$$

for $A, B, C \in \mathcal{P}(S)$.

We will use frequently to two-element field $\text{GF}(2)$ known as the 2-element Galois field, $\text{GF}(2) = \{0, 1\}$. Addition “ \oplus ” and multiplication “ \cdot ” in this field are defined by the following tables:

Set-Theoretical Preliminaries

2.1 Introduction

We begin with sets, sequences, collection of sets and set-theoretical operations. Closure systems and their connection to closure operators are presented as they facilitate an elegant presentation of future mathematical results.

Set partitions are very important to clusterings (which, in most cases, are partitions of sets of objects). This topic is discussed in the context of

relations and, especially, of equivalence relations.

A presentation of partially ordered sets and a detailed study of the partially ordered set of partitions provides the mathematical underpinnings

of families of clusterings.

The chapter concludes with a discussion of Galois connections that relate formal concept analysis to biclustering.

We assume that the reader is familiar with the notions of Cartesian

product of sets, relations, and function, that are part of many texts.

2.2 Sets and Set Operations

For a finite set S the number of elements of S is denoted by $|S|$. The empty set is denoted by \emptyset .

We write $x \in S$ to denote the fact that x is an element of the set S .

The usual symbols are used to denote set-theoretical operations: $A \cup B$ is

the union of the sets A and B , $A \cap B$ is the intersection of the sets A and

B , and $A - B$ is the difference of the sets A and B .

The *symmetric difference* of the sets A and B is denoted by $A \oplus B$. We

3.7	Entropy	109
3.7.1	Partition Entropy	112
3.7.2	Generalized Entropy	113
3.7.3	Conditional Entropy and Entropy Gain	115
3.7.4	The Entropic Metric Space of Partitions of Finite Sets	118
3.8	Exercises and Supplements	127
3.9	Bibliographical Comments	136
4.	Convexity	137
4.1	Introduction	137
4.2	Convex Sets	137
4.3	Operations on Convex Sets	155
4.4	Extreme Points	156
4.5	Convex Functions	160
4.6	Exercises and Supplements	168
4.7	Bibliographical Comments	174
5.	Graphs and Hypergraphs	175
5.1	Introduction	175
5.2	Vertices, Edges and Weights	175
5.3	Trees	195
5.3.1	Heaps	204
5.3.2	k - d -Trees	207
5.4	Bipartite Graphs	211
5.5	Co-cycles	228
5.6	$\text{GF}(2)$ -Linear Spaces and Graphs	233
5.7	Graphs and Matrices	238
5.8	Directed Graphs	243
5.9	From Matrices to Graphs	252
5.10	Minimum Spanning Trees	256
5.11	Matrices and Dissimilarities	264
5.12	Planar Graphs	265
5.13	Voronoi Diagrams	268
5.14	Graph Searching	270
5.15	Flows in Graphs	279
5.16	Hypergraphs	283
5.17	Exercises and Supplements	289

5.18	Bibliographical Comments	309
6.	Partitional Clustering	311
6.1	Introduction	311
6.2	Inertia of a Set of Vectors	311
6.3	The k -Means Algorithm	317
6.4	Clustering and Matrices	326
6.5	The PAM Algorithm	331
6.6	Kernel k -Means Clustering	334
6.7	A Geometric Approach to k -Clustering	339
6.8	Clustering and Singular Value Decomposition of Matrices	341
6.9	Vector Quantization and Matching Pursuit	343
6.10	Partitional Clustering in \mathbf{R}	348
6.11	Clustering in Python	355
6.12	Exercises and Supplements	359
6.13	Bibliographical Comments	387
7.	Statistical Approaches to Clustering	389
7.1	Introduction	389
7.2	Sampling	389
7.3	Likelihood Function	391
7.4	Data Density Estimations	394
7.5	Density Kernels for Unidimensional Data	396
7.6	Density Kernels for Multidimensional Data	402
7.7	Mean Shift Clustering	404
7.8	Expectation Maximization and Clustering	411
7.9	The EM Algorithm for Unidimensional Data	416
7.10	Exercises and Supplements	421
7.11	Bibliographical Comments	427
8.	Hierarchical Clustering	429
8.1	Introduction	429
8.2	Ultrametrics, Hierarchies, and Partition Chains	429
8.3	Ultrametrics and Minimum Spanning Trees	441
8.4	The Single-Link Algorithm	447
8.5	Other Hierarchical Clustering Algorithms	449
8.6	Hierarchical Clustering in \mathbf{R}	455
8.7	Hierarchical Clustering in Python	461

This page intentionally left blank

Machine learning algorithms struggle to match the human performance. Many clustering algorithms require the number of clusters to be provided as an input parameter, which forces these algorithms to combine or split natural clusters, or produce clusters that do not exist naturally in data. The pursuit of clusterings with a prescribed number of clusters is an ill-posed problem because a set of points can be clustered in many ways. Even if a data set has no meaningful structure, a clustering algorithm may find some partition of the data.

For the data set shown in Figure 1.1 a clustering algorithm that starts with a prescribed number of two clusters may split this data into two arbitrary clusters defined by the separating line ℓ (see Figure 1.3).

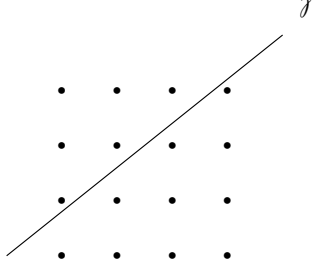


Fig. 1.3 Line separating the data set into two arbitrary clusters.

A clustering algorithm acting on the data set shown in Figure 1.2 may find two clusters or three clusters depending on the decision to fuse or not the two leftmost point groupings (which are very close). There are many types of clustering algorithms, many of them are covered in this text. The most important are:

- partitional algorithms (represented by the k -means algorithm and its variants);
- hierarchical algorithms (which include agglomerative and divisive algorithms);
- other classes include density-based clustering, grid-based clustering, spectral clustering;
- specialized algorithms have been developed for clustering categorical data, for stream data, for collections of documents and multimedia data, for time series, etc.

465	8.8	Divisive Hierarchical Clustering
470	8.9	The CURE Algorithm
476	8.10	Complexity of Hierarchical Clustering
484	8.11	Exercises and Supplements
491	8.12	Bibliographical Comments
493	9.	Density-based Clustering
493	9.1	Introduction
493	9.2	Core, Border and Noise Objects in DBSCAN
500	9.3	The OPTICS Algorithm
508	9.4	Density-based Clustering in R
512	9.5	Density-based Clustering in Python
515	9.6	Subspace Clustering
521	9.7	Exercises and Supplements
525	9.8	Bibliographical Comments
527	10.	Categorical Data Clustering
527	10.1	Introduction
528	10.2	Market Basket Data and Clustering Sets
531	10.3	The ROCK Algorithm
536	10.4	The CACTUS Approach
541	10.5	An Entropy-based Framework
545	10.6	Exercises and Supplements
548	10.7	Bibliographical Comments
549	11.	Spectral Clustering
549	11.1	Introduction
549	11.2	Data and Graphs
550	11.3	The Ordinary Spectrum of a Graph
552	11.4	The Laplacian Spectrum of a Graph
567	11.5	Cuts, Separators, and Clusterings
582	11.6	Spectral Clustering Algorithms
591	11.7	Spectral Clustering in R
594		Exercises and Supplements
603		Bibliographical Comments
605	12.	Correlation and Consensus Clustering
605	12.1	Introduction

12.2	Graphs and Correlation Clustering	606
12.3	The NP-Completeness of Correlation Clustering	607
12.4	Disagreements Minimization	609
12.5	Consensus Clustering	622
12.6	Exercises and Supplements	627
12.7	Bibliographical Comments	630
13.	Clustering Quality	631
13.1	Introduction	631
13.2	Internal Criteria	631
13.2.1	The Davies-Bouldin Criterion	631
13.2.2	The Dunn Quality Indices	634
13.2.3	The Silhouette Coefficient	634
13.3	External Criteria	637
13.4	Pairwise Measures	642
13.5	Graph Clustering and Modularity	645
13.6	Exercises and Supplements	649
13.7	Bibliographical Comments	652
14.	Clustering Axiomatization	653
14.1	Introduction	653
14.2	Clustering Functions	654
14.3	An Impossibility Result	655
14.4	Antichains of Partitions	658
14.5	Centroid-based Clustering and Consistency	660
14.6	Partitioning Functions	661
14.7	An Axiomatization of Clustering Quality Measures	663
14.8	Exercises and Supplements	665
15.	Biclustering	673
15.1	Introduction	673
15.2	Reorderable Matrices	674
15.3	A Boolean Approach to Binary Data Sets Biclustering	677
15.4	The Cheng and Church Algorithm	684
15.5	Numerical Aspects of Biclustering of Binary Data Sets	695
15.6	Exercises and Supplements	697
16.	Semi-supervised Clustering	701

ing, is very much work in progress. Various clustering algorithms applied to the same data set may produce distinct types of clusterings and no general principles to guide algorithm selection exist.

Studying clustering requires a broad spectrum of mathematical disciplines ranging from combinatorics, topology, linear algebra, optimization theory, etc. We strived to make the book as self contained as possible, including some preliminary chapters, as well as a number of appendices.

Treating clustering as an optimization problem is difficult because for each type of clustering there are objective function that fail to have optimal properties; additionally, most optimization problems are intractable and the users must contend with approximate algorithms.

The existence of clusters in data, that is, data clusterability is hard to formalize due to the variety in data distribution and the inadequacy of certain basic notions of clustering.

Humans are very good at identifying groupings of objects, at least in the case of uni-, bi-, or even tri-dimensional sets of objects. An examination of the data shown in Figure 1.1 shows that there is no obvious grouping of objects.

On other hand, the data shown in Figure 1.2 contains some “natural” groupings.

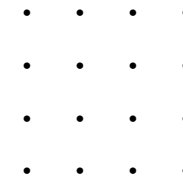


Fig. 1.1 Data set without an obvious grouping structure.

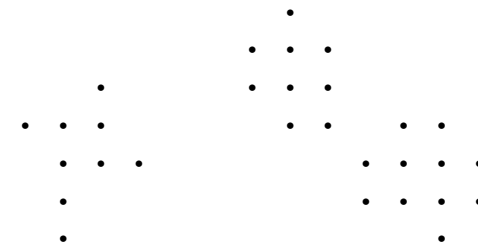


Fig. 1.2 Data that displays some grouping tendency.

Introduction

Clustering is the process of grouping a set of objects into subsets referred to as clusters according to some dissimilarity measure between objects. The goal is to group together similar objects, and to ensure that objects placed in distinct groups are dissimilar.

Supervised machine learning makes use of labelled data and creates a

model of the data that allows predicting the label for a yet unseen piece of data. In contrast, clustering belongs to the area of unsupervised machine learning defined as the task of discovering hidden structure from “unlabelled” data. Data items are not pre-categorized or labelled, which makes the evaluation of unsupervised learning algorithm difficult.

Supervised machine learning begins with a sample of data about which we have prior knowledge and tries to extrapolate this knowledge to larger volumes of data. Unsupervised learning, on the other hand, does not have prior knowledge, so its goal is to infer the “natural” structure present within

data.

Typical supervised learning activities include classification and regres-

sion; typical unsupervised activities include clustering and density estimation. Machine learning has been successful in the realm of supervised learning by providing significant understanding of various tasks, in constructing algorithmic tools to address them, insights about the alternative machine learning paradigms and their parameter settings, and initiating the development of new algorithmic approaches. No comparable successes are yet available in clustering which is a major unsupervised learning activity.

In general, unsupervised learning, utilizing the huge amounts of raw data available, is widely recognized as one of the most important challenges facing machine learning nowadays.

The unsupervised machine learning domain, and in particular cluster-

16.1	Introduction	701
16.2	A Semi-Supervised Variant of k -Means	702
16.3	Constraint Propagation	703
16.4	Semi-Supervised Hierarchical Clustering	706
16.5	Learning Mahalanobis Metrics	709
16.6	Exercises and Supplements	713
16.7	Bibliographical Comments	718
Appendix A Special Functions and Applications		
A.1	Introduction	719
A.2	Euler's Functions	719
A.3	Spheres and Cubes in \mathbb{R}^n	722
Appendix B Linear Algebra		
B.1	Introduction	727
B.2	Matrices	727
B.3	Matrix Rank	728
B.4	Matrix Differentiation	729
B.5	Eigenvalues of Matrices	734
B.6	Optimization and Eigenvalues	739
B.7	Matrix Norms	741
B.8	Unitary Matrices and Orthogonal Projections	747
B.9	Positive Definite Matrices	749
B.10	Singular Values of Matrices	751
B.11	CS Decomposition	757
B.12	Geometry of Subspaces	759
B.13	Matrix Approximation via SVD	761
Appendix C Linear Programming		
C.1	Introduction	763
C.2	The Fourier-Motzkin Elimination	763
C.3	Primal and Dual LP Problems	768
C.4	The Bin Packing Problem	774
C.5	Bibliographical Comments	775
Appendix D NP Completeness		
D.1	Introduction	777
D.2	Words and Problems	777

D.3	Problems and Algorithms	778
D.4	Variables and Propositional Formulas	780
D.5	Turing Machines	798
D.6	NP Complete Problems	799
D.7	Parameterized Problems	835
D.8	Approximation Algorithms	837
D.9	Bibliographical Comments	839
<i>Bibliography</i>		841
<i>Index</i>		855

This page intentionally left blank