

Spaces

Michael Betancourt

April 2023

Table of contents

1	Spaces	2
1.1	Sets	2
1.1.1	Variables	3
1.1.2	Subsets	3
1.2	Structures	8
1.2.1	Orderings	8
1.2.2	Algebras	9
1.2.3	Metrics	12
1.2.4	Topologies	14
2	Prototypical Spaces	25
2.1	Power Set	25
2.2	Integers	27
2.3	Real Lines	28
2.4	Extended Real Lines	29
3	Relating Spaces	30
3.1	Relating Sets	30
3.1.1	Classifying Functions	31
3.1.2	Function Inverses	36
3.1.3	Function Composition	37
3.2	Relating Structures	38
3.2.1	Pushforward and Pullback Relations	38
3.2.2	Bijjective Pushforwards and Pullbacks Relations	43
3.2.3	Structure-Preserving Relations	43
3.2.4	Reparameterizations	47
4	Conclusion	49

Acknowledgements	50
References	51
License	51

In [Chapter One](#) I introduced probability theory on finite sets; this simple setting allowed us to focus on the basic concepts without too much mathematical distraction. Unfortunately finite sets have only limited use in practical applications. Instead these applications typically require probability theory on not only more sophisticated sets but also sets that are equipped with additional structure. These combinations of sets and structure are also known as *spaces*.

In this chapter we will discuss the basic conceptual features of general spaces before reviewing some of prototypical spaces that are particularly common in practical applications. This presentation will include not only the properties of sets with an arbitrary number of elements but also a survey of some of the most fundamental structures that we can endow onto those sets.

1 Spaces

In both more applied and more pure mathematics the term “space” is omnipresent but subject to ambiguous and sometimes inconsistent interpretation. Some, for example, use the term to refer to any set of interest while some reserve it for sets endowed with additional mathematical *structure*. Still others use the term to denote sets endowed with not just any structure but particular, and often only implied, structure. Consequently we have to be careful when seeing the term to make sure we understand the exact intent of the author.

Throughout this book I will use the term **space** to describe *any* set equipped with *any* structure. When the particular structure is relevant I will be careful to specify exactly what structure has been endowed to the set. In this section we will review the properties shared by all sets and then some of the most common structures that we can introduce to enhance a set with additional complexity.

1.1 Sets

A finite set is a collection of a finite number of abstract **elements**. More generally a **set** is a collection of an *arbitrary* number of elements. The elements that compromise a set are also referred to as **points**, although this is often used only when the set is part of a space.

In general a set can contain not only a finite number of elements but also an *infinite* number of elements. An infinite set whose elements can be indexed by integers is referred as a **countably infinite set**. Finite and countably infinite sets are together denoted **countable sets**. When a set contains so many elements that they cannot be indexed by the integers then we refer

to it as an **uncountably infinite set** or simply an **uncountable set**. If we want to imply additional structure then we might also say **countable space** and **uncountable space**.

To work with a set in practice we will need to be able to reference the elements it contains and organize those elements into subsets.

1.1.1 Variables

Working with sets that contain more than a few elements is easier when we can refer to an abstract element without explicitly enumerating *all* of the elements. A **variable element**, or simply **variable**, is a symbol that can represent *any* element in a given set. In this book I maintain a convention where capital letters refer to sets and the corresponding lower case letters refer to variables. For example x denotes a variable taking values in the set X or, more compactly, $x \in X$.

To represent multiple, possibly distinct elements of a set we need to employ multiple variables. For example we might use ticks so that $x, x', x'' \in X$ all denote distinct variables that can correspond to common or different elements in X . We can also use integer indices to distinguish between these variables when helpful; in this case $x_1, x_2, x_3 \in X$ would denote three distinct variables that can correspond to common or different values in X . At the same time non-numerical labels such as $x_a, x_b, x_c \in X$ or even $x_{\text{input}}, x_{\text{output}} \in X$ can be more useful in some circumstances.

When two variables x_1 and x_2 represent the *same* element we say that they are *equal*. We can specify equality by writing $x_1 = x_2$.

In many circumstances we will need to distinguish between variables that refer to *arbitrary* elements and variables that refer to *particular* but unspecified elements. Following the computer science canon I will refer to these as **unbound variables** and **bound variables**, respectively. To distinguish between the two I will decorate bound variables with a tilde; in words x denotes any element of the space X while \tilde{x} denotes a fixed but unspecified element.

1.1.2 Subsets

As with finite spaces any selection of elements from a general set defines a **subset**. Subsets equipped with similar structure as the full set are also referred to as **subspaces**.

Selecting elements from a subset $x \subset X$ defines another subset $x' \subset X$. If x contains all of the elements of x' and more then we write $x' \subset x$; if it might contain only the elements in x' then we write $x' \subseteq x$. In the latter case we can also say that x' is *smaller* than x .

Following the treatment of finite spaces in the previous chapter I will refer to the collection of all subsets that can be selected from a set X as the corresponding **power set** and denote it 2^X . In general the power set 2^X will contain more elements than the originating set X ; the

power sets derived from uncountable sets are so massive that they are typically contaminated with all kinds of *misbehaving* subsets. We'll see one consequence of this contamination in the next chapter.

When a set contains only a small number of elements we can readily define each subset by *explicitly* specifying the elements it contains. For larger sets this quickly becomes impractical, if not outright impossible. In these cases subsets are more easily defined *implicitly* as the elements satisfying a certain condition. These implicit definitions are neatly encapsulated in the **set-builder notation**,

$$x = \{x \in X \mid \text{condition}(x)\}$$

which we can read as “the elements of X such that the element-wise condition is satisfied”.

For example if the function $f : X \rightarrow \mathbb{R}$ assigns to each element x a value $f(x)$ then we could define a subset as the collection of elements that are assigned the particular value 0. In set-builder notation this would become

$$x = \{x \in X \mid f(x) = 0\}.$$

The power set derived from any set always contains the **empty set** consisting of no elements, **atomic sets** or **singleton sets** each consisting of a single element, and a **full set** consisting of the entire set (Figure 1). Most subsets, however, contain an intermediate number of elements. One of the key features of uncountable spaces is that most subsets also contain an uncountable number of elements, but still only an infinitesimal proportion of the total set. To visually represent subsets containing an uncountable number of elements I will use filled shapes to contrast against individual points.

All of the set operations we introduced for finite sets generalize to any set. For example for any set X with power set 2^X we can define a unary operation that maps each subset into a **complementary** subset consisting of every other element (Figure 2),

$$\begin{aligned} \cdot^c : 2^X &\rightarrow 2^X \\ x &\mapsto x^c. \end{aligned}$$

For any set the complement of the empty set is the full set and vice versa,

$$\begin{aligned} \emptyset^c &= X \\ X^c &= \emptyset. \end{aligned}$$

The binary union and intersection operations also generalize to general sets (Figure 3). A **union** of two input subsets is the subset containing all of the elements contained in *either* input subset,

$$\begin{aligned} \cdot \cup \cdot : 2^X \times 2^X &\rightarrow 2^X \\ x_1, x_2 &\mapsto x_1 \cup x_2, \end{aligned}$$

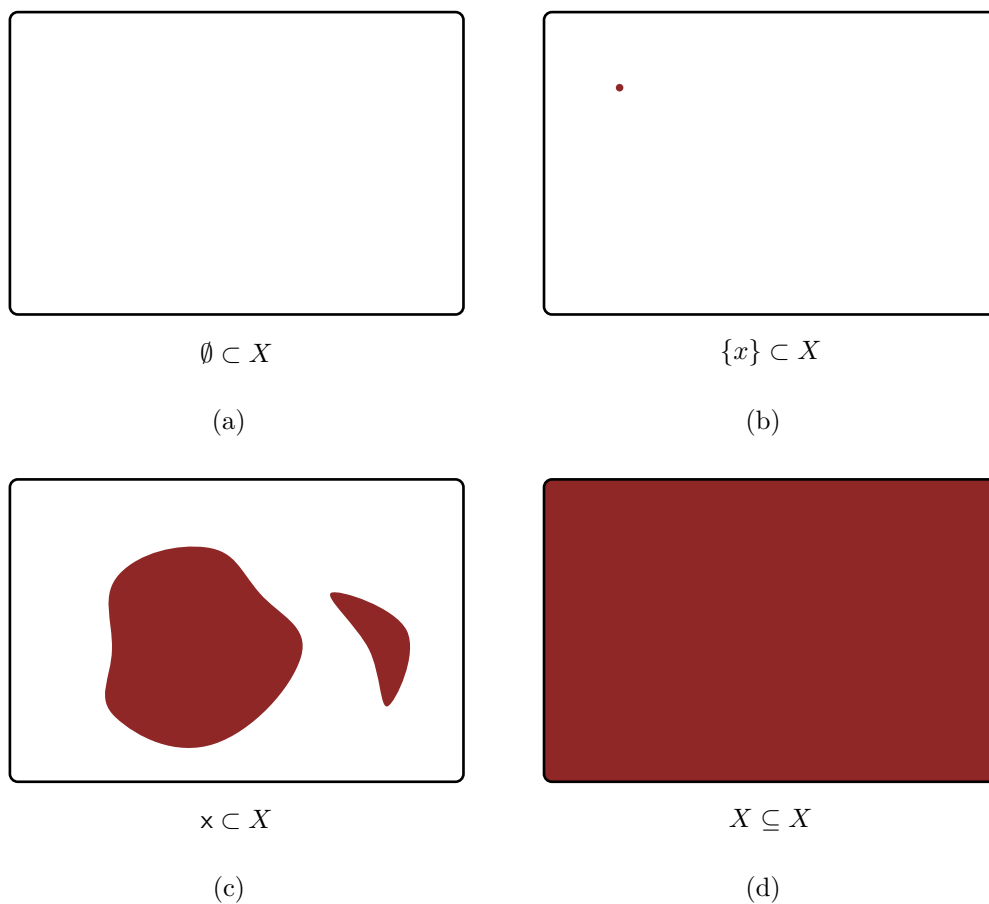


Figure 1: On a general set we can construct (a) an empty set containing no elements, (b) atomic sets containing only a single element, and even (d) a full set containing all elements in the set. (c) Most subsets are more intermediate, containing multiple elements but not all of them. Here filled shapes represent a potentially uncountable number of individual elements.

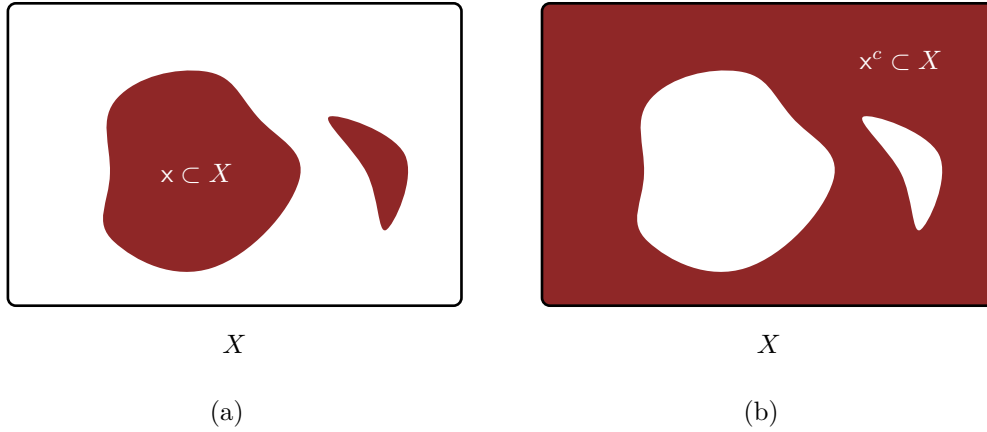


Figure 2: Every (a) subset defines a (b) complement subset consisting of all elements in the set that are not contained in that initial subset.

and the **intersection** is the subset containing all of the elements in *both* input subsets,

$$\begin{aligned} \cdot \cap \cdot : 2^X \times 2^X &\rightarrow 2^X \\ x_1, x_2 &\mapsto x_1 \cap x_2, \end{aligned}$$

Venn diagrams visualize unions and intersections as overlapping shapes.

Two subsets are said to be **disjoint** when they don't overlap and hence don't share any elements. More formally two subsets are disjoint when their intersection is the empty set,

$$x_1 \cap x_2 = \emptyset.$$

All of the properties of these binary operations that we discussed for finite sets also generalize to arbitrary sets. For example the union and intersection of a subset with itself always returns that subset,

$$x \cup x = x \cap x = x.$$

Similarly the union of the empty set with any subset returns back that subset,

$$x \cup \emptyset = \emptyset \cup x = x,$$

and intersection of the empty set with any subset returns back the empty set,

$$x \cap \emptyset = \emptyset \cap x = \emptyset.$$

The union of any subset with the full set returns the full set,

$$x \cup X = X \cup x = X,$$

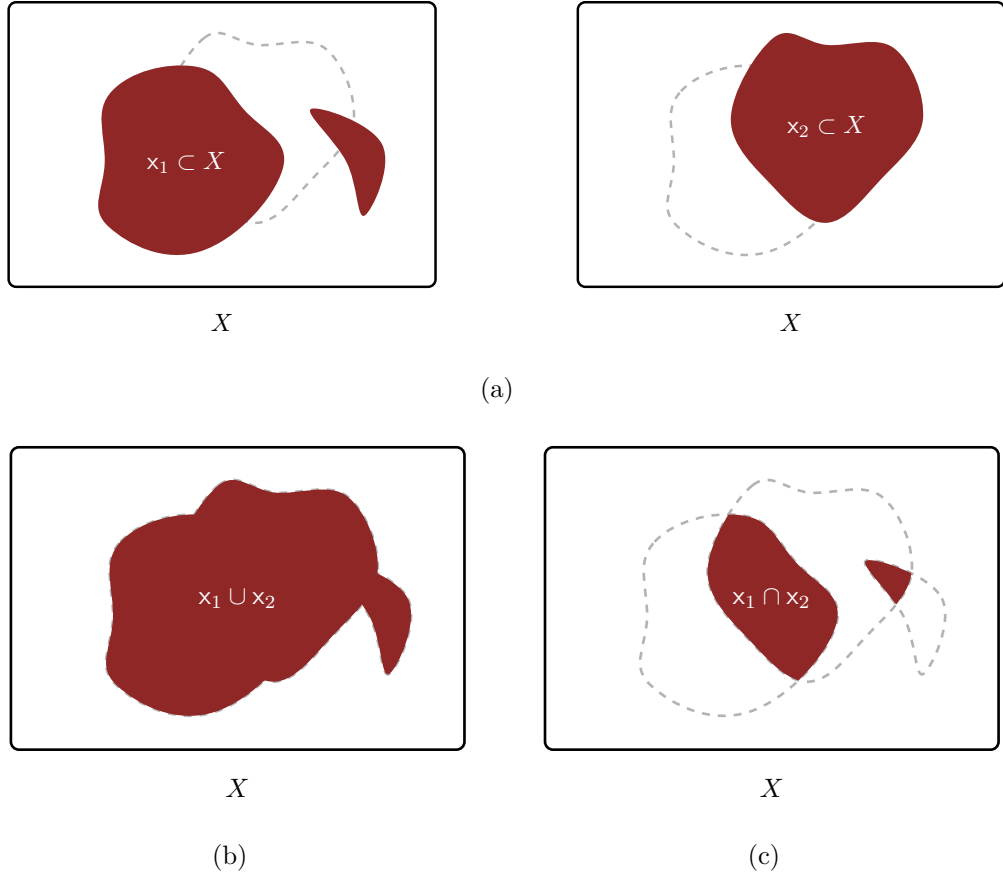


Figure 3: The union and intersection operations transform two input subsets into a single output subset. (a) A union consists of all elements in either input subset while (b) an intersection consists of all elements in both input subsets.

and the intersection of any subset with the full set returns back the subset,

$$\mathbf{x} \cap X = X \cap \mathbf{x} = \mathbf{x}.$$

The union and intersection operations also interact nicely with the complement operation. For example by construction the union of a subset and its complement is the full set,

$$\mathbf{x} \cup \mathbf{x}^c = X.$$

On the other hand the two subsets share no elements so their intersection is empty,

$$\mathbf{x} \cap \mathbf{x}^c = \emptyset.$$

Consequently a subset and its complement are always disjoint.

1.2 Structures

A collection of elements alone is only so interesting mathematically and only so useful in practice. The sets of theoretical and practical interest are equipped with additional **structures** that endow them with more rigidity and interpretability.

Earlier I defined a space as any a set equipped with any structure. When that structure is explicit we often denote the space by specifying the underlying set and the additional structure together. For example if we denote the set X and the structure \mathfrak{s} then we would denote the space (X, \mathfrak{s}) . Unfortunately these additional structures are often implicit and taken for granted. In this case X is, frustratingly, typically used to denote both the underlying set *and* the resulting space.

In this section we will review just a few of the structures that play important roles in practical applications. Specifically we will focus on the structures that define the most common spaces that arise in mathematical modeling. We will introduce probabilistic structures until the next chapter.

1.2.1 Orderings

One way to enhance a set is to distinguish a special organization of its elements.

An **strict ordering** or **complete ordering** establishes a particular sequential arrangement of the elements in a set. Mathematically we can encode an ordering as local relations between elements, writing $x_1 < x_2$ if the element x_1 appears earlier in the ordering than x_2 . We also say that x_1 is *smaller* than x_2 , and x_2 is *larger* than x_1 , if $x_1 < x_2$. The ordering relation can also be interpreted as a binary operation that takes two elements as inputs and returns true if the first input element appears earlier in the ordering than the second input element and false otherwise.

A consistent ordering of the elements in a set strongly constrains the behavior of these relations. For example if $x_1 < x_2$ and $x_2 < x_3$ then we must have $x_1 < x_3$. This property is formally known as **transitivity** of the individual relations.

A **partial ordering** defines a less rigid arrangement of a set that doesn't allow multiple elements to be indistinguishable; in other words multiple elements can occupy the same position along the sequential arrangement at the same time. To represent a partial ordering we use the relation \leq and write $x_1 \leq x_2$ if the element appears earlier in the order *or at the same place in the order as* x_2 . This partial ordering relation is not only transitive – if $x_1 \leq x_2$ and $x_2 \leq x_3$ then $x_1 \leq x_3$ – but also allows us to compare any element to itself – $x_1 \leq x_1$. This latter property is formally known as **reflexivity**.

Intuitively we can interpret a strict ordering as a geometric arrangement of the elements along a one-dimensional line. Partial orderings are more general and can often be interpreted as a geometrical arrangement of the elements along a higher-dimensional plane. Critically neither geometric interpretation relies on any notion of distance between the elements; any arrangement still leaves the individual elements quite fluid.

A set equipped with a particular ordering is referred to as an *ordered space*. Likewise a set equipped with a particular partial ordering is referred to as a *partially-ordered space*.

1.2.2 Algebras

The term **algebra** is introduced so early in our mathematical education that it can be especially easy to take for granted. Typically it refers to equipping a set with one or more *binary operations* that manipulate any two input elements into a single output element. Some algebraic structures also introduce *unary operations* that transform a single input element into a single output element.

Just about every possible algebraic structure has been studied and categorized by mathematicians. Unfortunately the resulting spaces are, with one exception, not referred to as “algebraic spaces”. In most cases they are not even referred to as “spaces”! Instead each algebraic structure is given a special name, such as “magma” and “group” and “field”, that is not obviously related to the others.

Because of this barrage of terminology it can be difficult to compare and contrast various algebraic structures. Fortunately understanding them all is absolutely not necessary for this book. For those readers who are interested in digger a little bit deeper into the technical vocabulary Wikipedia provides two nice reviews [here](#) and [here](#).

In this section we will consider only a few particularly common algebraic structures.

1.2.2.1 Single Algebraic Operation

A common convention when working with a single, binary algebraic operation is to denote that operation as

$$\begin{aligned}\cdot : X \times X &\rightarrow X \\ x_1, x_2 &\mapsto x_1 \cdot x_2,\end{aligned}$$

where x_1 denotes the first input element and x_2 denotes the second input element and $x_1 \cdot x_2$ denotes the output element. When working with multiple algebraic operations at the same time we will have to use multiple symbols to denote each operation.

Binary algebraic operations are characterized by a few key properties. For example a binary operation that is symmetric with respect to the input elements,

$$x_1 \cdot x_2 = x_2 \cdot x_1,$$

is referred to as a **commutative** operation.

A binary operation can be applied to more than two input elements by chaining multiple evaluations. For example we can apply a binary operation to the three elements x_1 , x_2 , and x_3 by first taking x_1 as x_2 inputs and then taking the resulting output as an input along with x_3 ,

$$(x_1 \cdot x_2) \cdot x_3.$$

Alternatively we can first take x_2 and x_3 as inputs and then take x_1 and the resulting output as inputs,

$$x_1 \cdot (x_2 \cdot x_3).$$

In general these two groupings will yield *different* outputs. Operations that always give the same output for both groupings are referred to as **associative** operations.

Some binary operations also distinguish special elements. For example inputting a particular element might reduce the operation to carrying forward the other input element to the output no matter what it might be. More formally a binary operation might admit an element x_{Id} such that

$$x \cdot x_{\text{Id}} = x_{\text{Id}} \cdot x = x.$$

for all $x \in X$. In this case x_{Id} is denoted a **unit element** or **identity element** and the binary operation itself is referred to as a **unital** operation.

Similarly an operation might allow a particular element to determine the output no matter what the other input element is. Formally there might exist an element x_{Null} such that

$$x \cdot x_{\text{Null}} = x_{\text{Null}} \cdot x = x_{\text{Null}}$$

for all $x \in X$. If so then x_{Null} can be denoted an **absorbing element**, **null element**, or **zero element** with the binary operation sometimes referred to as a **null** operation.

The existence of an identity element can also induce a special pairing between the other elements in the set. In particular some elements $x \in X$ might be paired with another element $x^{-1} \in X$ such that the binary operation returns the identity element,

$$x \cdot x^{-1} = x^{-1} \cdot x = x_{\text{Id}}.$$

If so then we say that x^{-1} is the **inverse element** of x . By construction x is also the inverse element of x^{-1} . By construction the identity element is always its own inverse element. Conversely, because null elements cannot result in an identity element output, they will never have inverse elements.

Perhaps the most prominent binary algebraic operation is **addition** over the integers and real numbers. Addition is commutative, associative, and unital with the identity element $x_{\text{Id}} = 0$. Every element x also features a unique additive inverse element known as its **negation** and written as $-x$. A corresponding **subtraction** operation can be defined as addition by additive inverses,

$$x_1 - x_2 \equiv x_1 + (x_2)^{-1} = x_1 + (-x_2).$$

A natural complement to addition is **multiplication**. Multiplication is commutative, associative, and unital with the identity element $x_{\text{Id}} = 1$. It is also null with the absorbing element $x_{\text{Null}} = 0$.

When working with the integers no element admits a multiplicative inverse except for the identity element. On the other hand every real number x except for $x_{\text{Null}} = 0$ features a multiplicative inverse element known as its **reciprocal** and written as $1/x$. We can define a **division** operation as multiplication by multiplicative inverses,

$$x_1/x_2 \equiv x_1 \cdot (x_2)^{-1} = x_1 \cdot \frac{1}{x_2}.$$

Because $x_{\text{Null}} = 0$ doesn't have a multiplicative inverse, however, we cannot divide by zero.

1.2.2.2 Multiple Algebraic Operations

If one algebraic operation is good then two should be better, right? Multiple algebraic operations that are *compatible* with each other define particularly rich algebraic structures.

In the context of two binary algebraic operation a common convention is to write the first operator as

$$\begin{aligned} + : X \times X &\rightarrow X \\ x_1, x_2 &\mapsto x_1 + x_2 \end{aligned}$$

and the second as

$$\begin{aligned} \cdot : X \times X &\rightarrow X \\ x_1, x_2 &\mapsto x_1 \cdot x_2 \end{aligned}$$

in analogy with addition and multiplication. This notation might be used even if the operators do not actually correspond to the usual addition and multiplication operations.

The binary operation \cdot is said to **distribute** across the binary operation $+$ if

$$x_3 \cdot (x_1 + x_2) = (x_3 \cdot x_1) + (x_3 \cdot x_2).$$

Note that this property is asymmetric – it does *not* imply that $+$ distributes across \cdot ,

$$x_3 + (x_1 \cdot x_2) \neq (x_3 + x_1) \cdot (x_3 + x_2).$$

A common example of compatible algebraic operations is addition and multiplication over the integer and real numbers where multiplication distributes across addition. An algebraic structure comprised of these two compatible operations is also known as **arithmetic**.

One of the most infamous algebraic structures considers a slightly different combination of operations. Instead of a binary operator that takes two elements as inputs we can define a binary operator that takes as inputs one element and one real number,

$$\begin{aligned} \cdot : \mathbb{R} \times X &\rightarrow X \\ \alpha, x &\mapsto \alpha \cdot x. \end{aligned}$$

This is sometimes referred to as a **scalar multiplication** operator.

A **linear algebra** combines this operation with an associative, commutative, and unital binary operator

$$\begin{aligned} + : X \times X &\rightarrow X \\ x_1, x_2 &\mapsto x_1 + x_2 \end{aligned}$$

such that \cdot distributes over $+$,

$$\alpha \cdot (x_1 + x_2) = \alpha \cdot x_1 + \alpha \cdot x_2.$$

A set equipped with these operators is also known as a **vector space** with the individual elements denoted **vectors**.

1.2.3 Metrics

Another way to add structure to a set is to introduce an explicit notion of **distance** between individual elements. Mathematically this is done with a binary operation that takes two elements as inputs and returns a positive real number as output,

$$\begin{aligned} d : X \times X &\rightarrow \mathbb{R}^+ \\ x_1, x_2 &\mapsto d(x_1, x_2). \end{aligned}$$

In order for this operation to conform with our intuitive notions of distance we need it to satisfy a few key properties. For example the output should vanish when comparing an element to itself,

$$d(x_1, x_2) = 0$$

when $x_1 = x_2$. At the same time the output should be non-zero when comparing two distinct inputs,

$$d(x_1, x_2) > 0$$

when $x_1 \neq x_2$. The operation should also be symmetric in its inputs,

$$d(x_1, x_2) = d(x_2, x_1),$$

and for any three elements satisfy a *triangle inequality*,

$$d(x_1, x_3) \leq d(x_1, x_2) + d(x_2, x_3).$$

If the operation satisfies all of these properties then we can consistently say that the larger $d(x_1, x_2)$ is the most distant x_1 is from x_2 .

Any operation satisfying these criteria is known as a **distance function** or **metric**. A set equipped with a metric is denoted a **metric space**.

Because a metric is symmetric in its inputs it is insensitive to any ordering of the elements. In fact the distances defined by a metric complement the arrangement defined by an ordering: while an ordering defines how elements are arranged but not how far apart they are, a metric defines how far apart the elements are but not their arrangement. Together an ordering and metric define a rigid arrangement of the elements.

One of the most important applications of a metric is formalizing notions of *convergence* and *limits* in a set. Consider for example a countably infinite sequence of elements

$$\{x_1, x_2, \dots, x_i, \dots\}.$$

A metric quantifies how far these elements are from any other element $x \in X$,

$$\{r_1, r_2, \dots, r_i, \dots\} = \{d(x_1, x), d(x_2, x), \dots, d(x_i, x)\}.$$

Intuitively if these distances decay towards zero then the sequence of elements will move closer and closer to x .

More formally a sequence of elements **converges** to the **limit** x if for *any* distance $\epsilon > 0$ we can go deep enough into the sequence that the distances between the tail elements

$$\{x_{i'}, x_{i'+1}, \dots, x_{i'+i}, \dots\}$$

and the limiting element x are all bounded by ϵ ,

$$\begin{aligned} d(x_{i'}, x) &< \epsilon \\ d(x_{i'+1}, x) &< \epsilon \\ &\dots \\ d(x_{i'+i}, x) &< \epsilon \\ &\dots, \end{aligned}$$

no matter how small it might be (Figure 4). In other words the elements of a convergent sequence will become arbitrarily close to x , at least eventually.

Convergent sequences can then be used to formally implement limits. In particular the notation $x' \rightarrow x$ corresponds to *any* sequence of elements that converges to x . Limiting behavior is determined by how *all* of these convergent sequences behave.

1.2.4 Topologies

The distances defined by a metric endow a set with a rich diversity of useful features. Many of these features, however, can also be defined by distinguishing certain subsets, avoiding the need for a metric entirely.

1.2.4.1 Open Balls

A metric can be used to construct special subsets. The collection of all elements within a distance r from the element $x \in X$ defines a subset

$$\mathbf{b}_r(x) = \{x' \in X \mid d(x, x') < r\} \subseteq 2^X$$

known as an **open ball** centered at x with radius r . Any open ball with zero radius reduces to the empty set.

Including the elements exactly at a distance r from x defines a slightly larger subset

$$\bar{\mathbf{b}}_r(x) = \{x' \in X \mid d(x, x') \leq r\} \subseteq 2^X$$

known as a **closed ball** centered at x with radius r . By themselves the elements exactly a distance of r away from x define yet another subset

$$\delta\mathbf{b}_r(x) = \{x' \in X \mid d(x, x') = r\} \subseteq 2^X$$

that defines the **boundary** of the open and closed balls. An open ball becomes a closed ball when we combine it with this boundary,

$$\bar{\mathbf{b}}_r(x) = \mathbf{b}_r(x) \cup \delta\mathbf{b}_r(x).$$

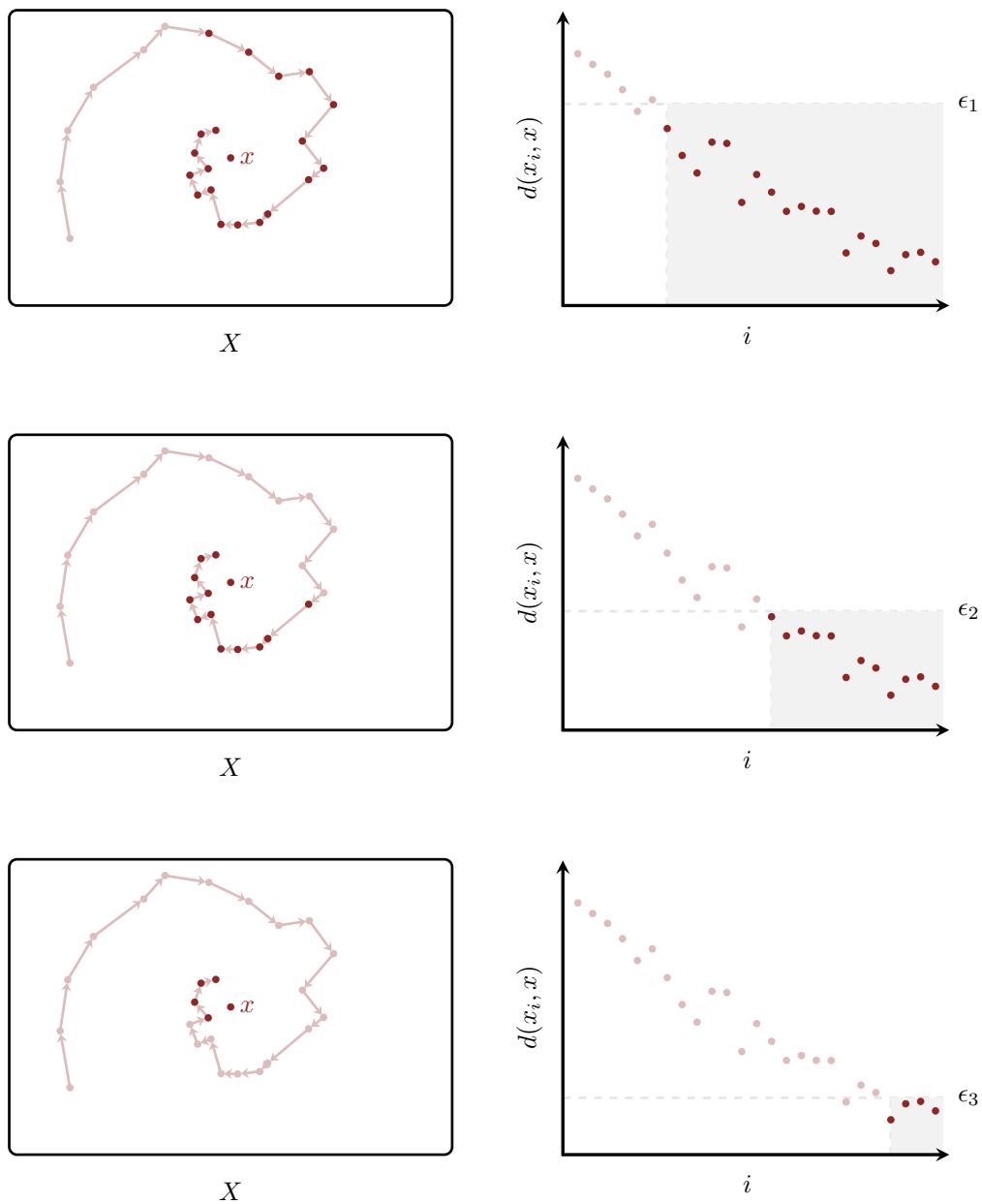


Figure 4: Equipping a set with a metric allows us to quantify when a sequence of elements converges to a limiting element $x \in X$. If the distance between the sequential elements and the limiting element x is eventually bounded by any finite distance ϵ then the sequential elements will become arbitrarily close to x .

Open balls quantify which elements are in the *proximity* of the central element without directly referencing the metric. More formally every open ball centered at $x \in X$ with finite but potentially arbitrarily small radius defines a **metric neighborhood** of x . The smaller a metric neighborhood is the closer the elements it contains will be to the central element x . To emphasize the possibility of arbitrarily small metric neighborhoods I will use ϵ instead of r when referring to the open ball radii that define metric neighborhoods.

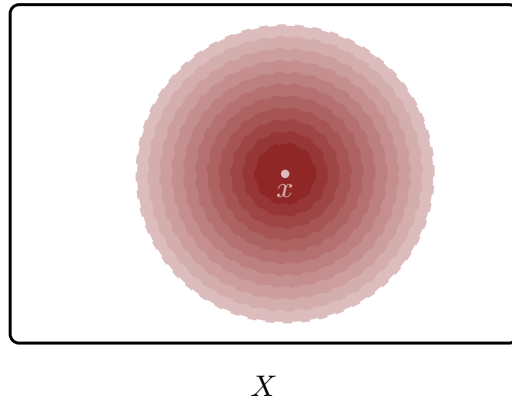


Figure 5: Open balls centered at $x \in X$ define neighborhoods of elements that are close to x . The smaller the neighborhood the closer the elements are to x .

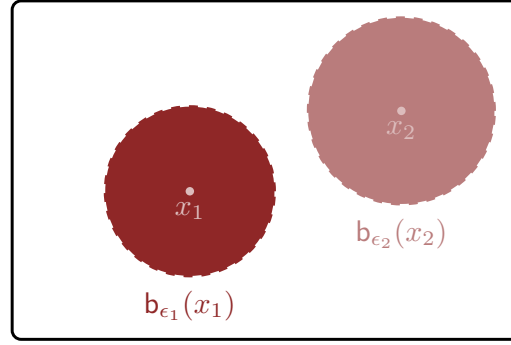
The interaction of these neighborhoods reveals quite a bit about the relationships between individual elements in a metric space. For example because any pair of distinct elements are separated by a finite distance they can always be contained in non-overlapping metric neighborhoods (Figure 6). In other words metric neighborhoods are able to *distinguish* individual elements from each other.

Moreover we can use metric neighborhoods to define convergent sequences. Recall that, by definition, we can always go deep enough into a convergent sequence that the tail elements

$$\{x_{i'}, x_{i'+1}, \dots, x_{i'+i}, \dots\}$$

will be closer to the limiting element x than any distance $\epsilon > 0$,

$$\begin{aligned} d(x_{i'}, x) &< \epsilon \\ d(x_{i'+1}, x) &< \epsilon \\ &\dots \\ d(x_{i'+i}, x) &< \epsilon \\ &\dots, \end{aligned}$$



X

Figure 6: For any two elements $x_1, x_2 \in X$ in a metric space we can always find radii $\epsilon_1 + \epsilon_2 < d(x_1, x_2)$ such that the open balls $b_{\epsilon_1}(x_1)$ and $b_{\epsilon_2}(x_2)$ containing the two elements will not overlap. Consequently metric neighborhoods are always able to separate individual elements.

This, however, is equivalent to requiring that those tail elements are contained in the open ball $b_\epsilon(x)$ centered at the limiting element,

$$\{x_{i'}, x_{i'+1}, \dots, x_{i'+i}, \dots\} \in b_\epsilon(x).$$

In other words a sequence will converge to the limit x if and only if the elements in the sequence are eventually contained in *any* metric neighborhood of x (Figure 7).

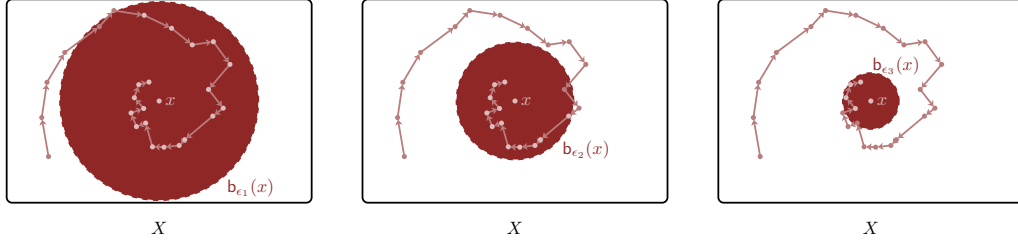


Figure 7: On a metric space a sequence of elements converges to the limit $x \in X$ if and only if we can always go deep enough in the sequence that all subsequent elements will be contained in any metric neighborhood.

Metric neighborhoods can also be used to formalize intuitive notions of *continuous* and *discrete* metric spaces. For example we can formalize the notion of a continuum by requiring that every element $x \in X$ is surrounded by elements at any arbitrary distance, at least up to some potentially maximum distance. More succinctly for *every* element $x \in X$ and distance

$0 < \epsilon \leq \epsilon_{\max}$ we should be able to find another element $x' \in X$ satisfying

$$d(x, x') = \epsilon.$$

This requirement, however, is equivalent to every metric neighborhood with radius less than ϵ_{\max} defining a *distinct* subset of X because increasing the radius will always introduce new elements. This then implies that every metric neighborhood of x , no matter how small, must contain more elements than just x . In other words no metric neighborhood in such a space will reduce to an atomic set.

On the other hand the elements in a discrete metric space are not separated by arbitrary distances. Instead they are separated only by certain distances, specifically distances bounded below by some minimum distance. In this case many metric neighborhoods are degenerate, reducing to the same subsets of X (Figure 8). Moreover any metric neighborhood with radius smaller than the minimum distance will reduce to the atomic set containing only the central element. These smallest neighborhoods isolate the elements from each other.

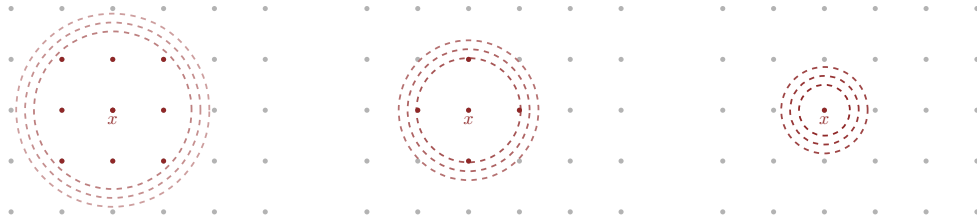


Figure 8: The elements in a discrete metric space are separated by some finite minimum distance, which is equivalent to many metric neighborhoods degenerating into the same subsets. In particular the smallest metric neighborhoods all degenerate to the atomic set which contains only the central element.

1.2.4.2 Open Sets

The concept of an open subsets can also be generalized beyond just open balls. Consider for example a subset $x \subset X$ and an element $x \in x$. If x is contained in a metric neighborhood that is itself contained in the subset,

$$x \in \mathbf{b}_\epsilon(x) \subset x,$$

then we can move away from x without ever leaving x . When *every* element in x can be contained by at least one metric neighborhood that is also contained in x then we can always move closer to the edge of the subset without leaving it.

A key consequence of this construction is that these subsets, just like the open balls, do not contain any boundary elements. Consequently they make a natural generalization for **open**

subsets (Figure 9). We can then define **closed subsets** as the complements of any open subset. Closed subsets are characterized by their inclusion of elements whose metric neighborhoods all leak out into the surrounding set (Figure 10).

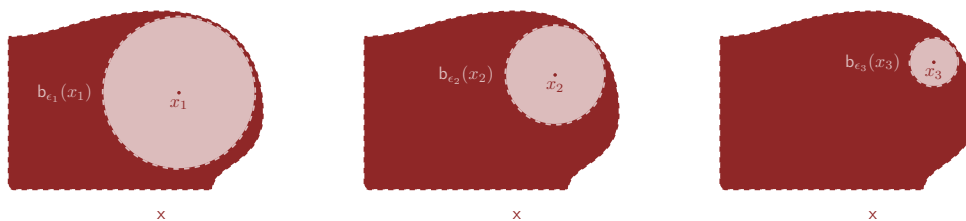


Figure 9: In a metric space every element of an open subset can be contained by a metric neighborhood that is itself fully contained by the open subset. As we move closer and closer to the boundary of an open subset the metric neighborhood will become smaller but never vanish, allowing us to make progressively smaller moves towards the boundary without ever reaching it.

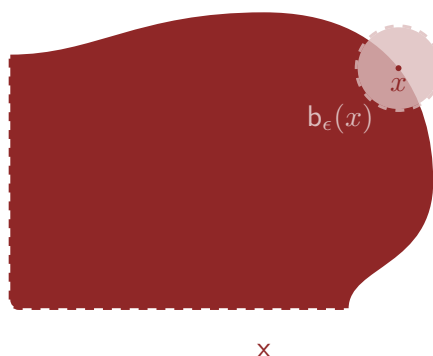


Figure 10: A closed subset of a metric space contains boundary elements whose metric neighborhoods always leak out of the subset.

Unfortunately this construction does introduce a few subtleties. Unlike the colloquial use of “open” and “closed” these definitions of open subsets and closed subsets are not exact opposites. If a subset and its complement are both open then they will, by definition, be both open and closed at the same time. Because mathematicians have no shame these subsets are referred to as “clopen” subsets. Moreover open subsets and closed subsets are both exceptional in the power set; in general only *some* subsets will be open, closed, or both, and most subsets will be neither open nor closed.

Unlike open balls these metric-derived open subsets are *closed* under unions and intersections. If x_1 and x_2 are both open subsets then $x_1 \cup x_2$ will also be an open subset. In fact the union

of any *countably infinite* number of open subsets will be open. Likewise if x_1 and x_2 are both open subsets then $x_1 \cap x_2$ will also be an open subset. Indeed the intersection of any *finite* number of open subsets will also be open but, in an unfortunate asymmetry, the intersection of a countably infinite number of open subsets might not be open.

Open subsets provide an immediate way to generalize the notion of neighborhood beyond open balls. If we define a general neighborhood of x to be *any* open subset that contains x then we will be able to derive many of the same properties as metric neighborhoods, and hence the underlying metric. For example two elements can be separated by metric neighborhoods if and only if they can be separated by open subsets more generally. Similarly every sequence that converges to the limit x is eventually contained not just by every metric neighborhood of x but also by every general neighborhood of x .

While different metrics define different metric neighborhoods around each element, those different metric neighborhoods can sometimes end up defining the *same* open subsets (Figure 11). Because those open subsets completely characterize properties like separation and convergence the metrics that yield the same open subsets will all share those properties! Working directly with a collection of open subsets allows us to avoid the idiosyncrasies of any particular metric and isolate the properties common to *all* compatible metrics.

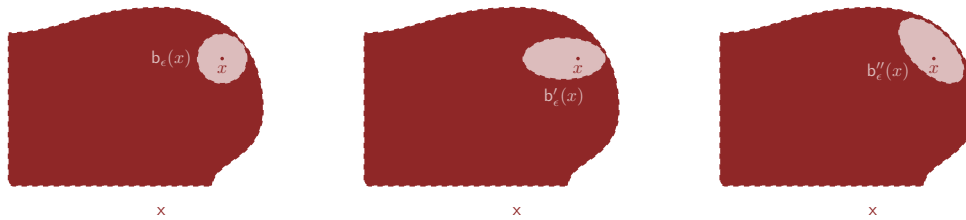


Figure 11: Different metrics define different metric neighborhoods, but in some cases those different metric neighborhoods end up defining the same collection of open subsets. Any property defined by those common open subsets will be shared by all of the compatible metrics.

1.2.4.3 General Topologies

Once we can define interesting properties directly from open subsets a natural question is whether or not those open subsets need to be derived from any existing structure in the first place. Indeed we can introduce all kinds of interesting properties to a set by equipping it directly with a self-contained collection of distinguished subsets.

A **topology** is any collection of subsets

$$\mathbf{t} = \{\emptyset, x_1, \dots, x_i, \dots, X\} \in 2^X$$

that contains the empty set and the full set, is closed under countably infinite unions, and finite intersections. Any set equipped with a topology is referred to as a **topological space**.

The distinguished subsets that define a topology are generally referred to as **open subsets** even if they are not derived from a metric. If the open subsets in a topology are derived from a metric, or *can be* derived from a metric, then we refer to the topology as a **metric topology**.

Non-empty subsets in a topology provide the ultimate generalization of neighborhoods. Every subset in a topology that contains the element $x \in X$ defines a neighborhood of x , with each neighborhood providing some quantification of which elements in the set are in the proximity of x (Figure 12). Smaller neighborhoods imply a stronger sense of proximity than larger neighborhoods.

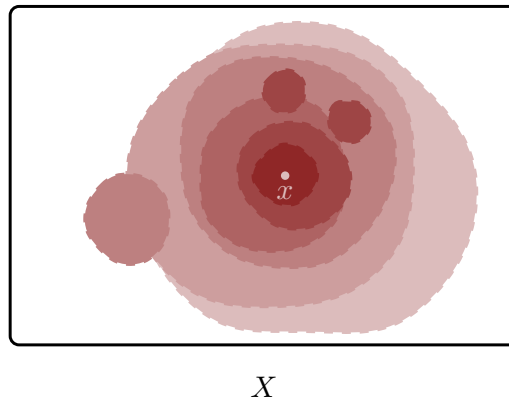


Figure 12: The subsets in a topology define a general notion of neighborhood. Any subset in the topology that contains an element $x \in X$ defines a neighborhood of elements in the proximity of x .

The overlap of topological neighborhoods around distinct elements determines how well the topology can distinguish those elements from each other. Larger topologies contain more neighborhoods that better resolve the individual elements in the underlying set than smaller topologies. In other words the larger the topology the closer we can zoom into the individual elements.

For example the smallest topology, known as the **trivial topology**, contains only the empty set and the full set,

$$\tau = \{\emptyset, X\}.$$

In this case there is only one neighborhood, and that neighborhood is shared by *all* of the elements in X . Consequently the topology is not able to discriminate any element from any other element.

At the other extreme the **discrete topology** contains the entire power set,

$$\tau = 2^X.$$

Because the discrete topology contains every atomic set each element can be contained in a neighborhood by itself, allowing us to zoom into each element individually and isolate it from every other element in X (Figure 13). This generalizes the behavior of discrete metric spaces that we encountered in Section 1.2.4.1.

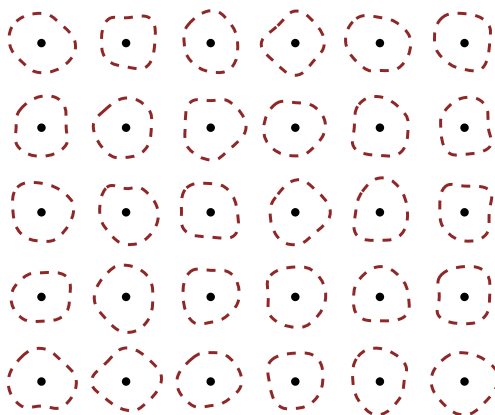


Figure 13: The discrete topology contains all of the atomic sets which define neighborhoods that completely isolate each element from every other element.

Particularly well-behaved topologies include enough subsets that every pair of elements can be contained by non-overlapping neighborhoods (Figure 14), mirroring the separation afforded by open subsets on metric spaces. In other words these **Hausdorff topologies** are able to distinguish every element from every other element.

The continuous metric spaces discussed in Section 1.2.4.1 are generalized by topologies that fall in between the extreme trivial and discrete topologies. They contain enough subsets to be Hausdorff but not so many to be discrete, allowing us to zoom in arbitrary close to each element but never isolate them completely from neighboring elements. The precise subsets included in these intermediate topologies then endow a set a certain notions of “shape”.

For example the half-open line interval

$$[0, 1) = \{x \in \mathbb{R} \mid 0 \leq x < 1\}$$

is a Hausdorff topological space with open subsets defined by half-open subintervals

$$[a, b) = \{x \in \mathbb{R} \mid 0 \leq a \leq x < b < 1\}$$

and their unions. In particular the half-open subintervals

$$[0, \epsilon) = \{x \in \mathbb{R} \mid 0 \leq x < \epsilon < 1\}$$

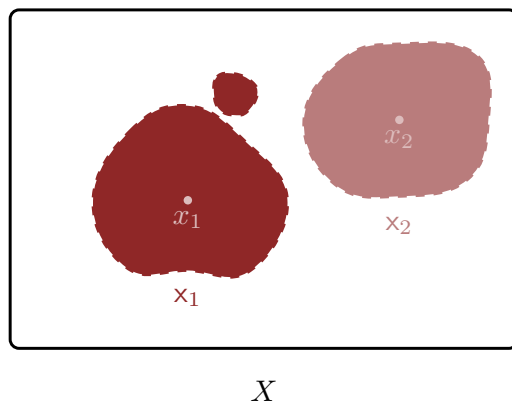


Figure 14: A Hausdorff topology includes enough subsets that every pair of elements $x_1, x_2 \in X$ can be contained by at least one set of neighborhoods that don't overlap with each other. More formally $x_1 \in \mathfrak{x}_1 \subset X$ and $x_2 \in \mathfrak{x}_2 \subset X$ with $\mathfrak{x}_1 \cap \mathfrak{x}_2 = \emptyset$.

contain elements near lower boundary but not elements near the upper boundary, separating the two ends from each other and giving the line its distinctive shape (Figure 15a). Removing these subintervals gives *another* Hausdorff topology, but one where the obstructions between the elements near the lower boundary and the elements near the upper boundary have been removed. This effectively connects the two ends together, endowing the set not with the shape of a line but rather the shape of a circle (Figure 15b)!

Convergence on topological spaces can also be defined in the same way as it is on metric spaces, replacing the neighborhoods derived from open balls to the neighborhoods defined by the topology (Figure 16). The generality of topological structure, however, does come at a cost: if a topology cannot discriminate between individual elements than the limit of a convergent sequence might not be unique. Instead unique limits are guaranteed by only Hausdorff topologies.

A subtle benefit of this topological definition of convergence is that because it doesn't require a metric it also doesn't require us to define the positive real numbers. This can be helpful for avoiding circular logic in more technical mathematical analyses.

1.2.4.4 The Practical Uses of Topology

In general topologies can be used to equip a set with all kinds of complicated structure, defining sophisticated spaces that far exceed our intuitions. Indeed the construction and classification of elaborate topological spaces is a major focus of more abstract mathematics. Because most introductions to topology focus on these objectives the subject has developed an imposing reputation of inaccessibility, if not outright irrelevance, to more applied aspirations.

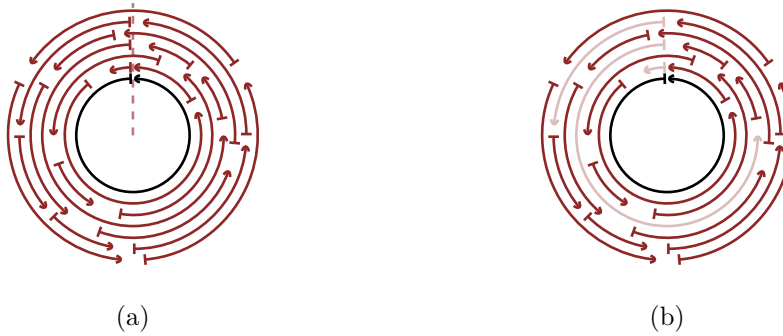


Figure 15: The precise subsets included in a topology endow the underlying set with subtle notions of shape. (a) A line is an uncountably infinite set equipped with a topology that contains enough subsets to strongly separate the ends from each other. (b) Removing those particular subsets, shown here in light red, gives a smaller topology where the ends are in closer proximity, giving the same set the shape of a circle.

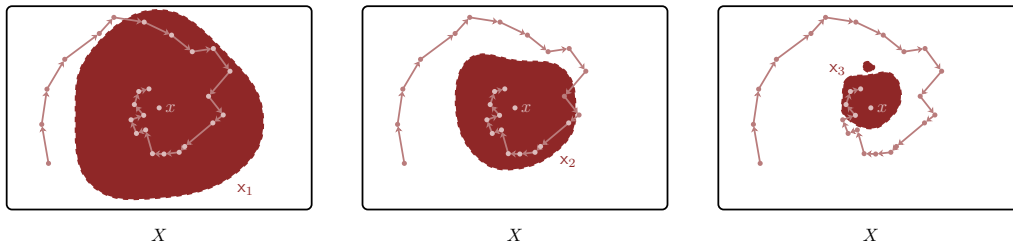


Figure 16: The subsets in a topology can be used to define convergent sequences in the same way that the open subsets in a metric space can: a sequence of elements converges to the limit $x \in X$ if and only if we can always go deep enough in the sequence that all subsequent elements can be contained in any topological neighborhood of x . For many topologies, however, a sequence can converge to multiple elements at the same time. Unique limits are a feature of only some topologies.

Topology, however, can be a powerful tool in practice by helping us to better understand the familiar spaces that appear in more applied mathematics. Specifically a given topology can be used to quantify which behaviors will be *shared* by different choices of other structures, such as orderings and metrics, and which behaviors are particular to those choices.

For example if a topology is not Hausdorff then we won't be able to distinguish between individual elements well enough to consistently define a metric that assigns non-zero distances $d(x_1, x_2) > 0$ to every pair of distinct elements $x_1 \neq x_2$. In other words *every* well-defined metric space must also be a Hausdorff topological space. Consequently *every* metric space shares the properties that can be derived from Hausdorff topological structure alone, such as separation of elements by neighborhoods and uniqueness of convergent sequence limits.

Some metric spaces have even more topological similarities. In [Section 1.2.4.2](#) we saw that different metrics on the same set can define the same open sets, and hence the same metric topology. Different metric spaces that induce the same metric topology are particularly alike, sharing every property that can be defined by that common topology. For example on metric spaces with the same topology convergent sequences will not only enjoy unique limits, they will share the *same* limits.

Similarly all spaces equipped with the topology of a line that we explored in [Section 1.2.4.3](#) will share the same basic linear “shape” regardless of any compatible metric structure we introduce, while all spaces equipped with the topology of a circle will exhibit the same circular “shape”. Different metrics can warp how far apart the elements are from each other but they cannot disrupt those topologically-derived shapes.

In other words working with topological properties allows us to isolate all of the behaviors shared by spaces equipped with compatible structures and disregard the idiosyncrasies any particular choice of that structure. Because of this topological structure is often interpreted as more *fundamental* than other structures. From this perspective the foundation of a space is a set equipped with a particular topology. Any additional structure, such as an ordering, algebra, or metric just builds on top of that foundation.

2 Prototypical Spaces

The more structure we endow onto a set the richer its properties will be. That richness offers not only mathematical but also interpretational opportunities: indeed the spaces that are most common in practice are absolutely spoiled with structure. In section we'll review the spaces that arise most often in mathematical modeling.

2.1 Power Set

The subsets of any set naturally feature a surprising amount of structure that we can use to elevate the power set into a space.

For example subsets can be partially ordered by inclusion. A subset $x_1 \subset X$ is smaller than a subset $x_2 \subset X$ if $x_1 \subset x_2$, and larger if $x_2 \subset x_1$. Two subsets that only partially overlap fall into the same place in the ordering. For any set the empty set will always be the smallest subset and the full set will always be the largest.

The union and intersection operations introduce algebraic structure, known as a **Boolean algebraic structure**, to the power set. They are both commutative,

$$\begin{aligned}x_1 \cup x_2 &= x_2 \cup x_1 \\x_1 \cap x_2 &= x_2 \cap x_1,\end{aligned}$$

and associative,

$$\begin{aligned}(x_1 \cup x_2) \cup x_3 &= x_1 \cup (x_2 \cup x_3) \\(x_1 \cap x_2) \cap x_3 &= x_1 \cap (x_2 \cap x_3).\end{aligned}$$

Both operations are also unital but with different identity elements. For example the identity element for the union operation is empty set,

$$x \cup \emptyset = \emptyset \cup x = x,$$

while the identity element for the intersection operation is the full set,

$$x \cap X = X \cap x = x.$$

Interestingly no elements but these identity elements admit inverse elements; no union can reduce a non-empty input subset to the empty set and no intersection can elevate a non-full input subset to the full set.

At the same time the union and intersection are null operators. Because the union of any subset with the full set returns the full set,

$$x \cup X = X \cup x = X,$$

the full set is an absorbing element. Similarly because the intersection of any subset with the empty set is the empty set,

$$x \cap \emptyset = \emptyset \cap x = \emptyset,$$

the empty set is an absorbing element.

These operations are also compatible with each other – not only do unions distribute across intersections,

$$x_3 \cup (x_1 \cap x_2) = (x_3 \cup x_1) \cap (x_3 \cup x_2),$$

but also intersections distribute across unions,

$$x_3 \cap (x_1 \cup x_2) = (x_3 \cap x_1) \cup (x_3 \cap x_2).$$

We can equip a power set with metrics and topologies, but the power set itself does not motivate any particular choice of these structures on its own. That said metrics and topologies endowed on the underlying set can be used to derive compatible metrics and topologies on the power set.

2.2 Integers

The **integers**, denoted \mathbb{Z} , are the prototypical discrete space which makes them particularly useful for modeling discrete quantities and indexing sequences of objects. That said despite their ubiquity we often take their defining structures for granted.

Formally the integers are built from a countably infinite set of elements equipped with a strict ordering, algebra, metric, and topology. The algebraic structure is given by the familiar arithmetic operations of addition and multiplication that are compatible with the ordering. We can also define additive inverses, and hence subtraction, but we cannot define division because the multiplicative inverses don't exist unless we expand the space to include fractions.

This algebraic structure can also be used to define a metric as

$$d(x_1, x_2) = \sqrt{(x_1 - x_2) \cdot (x_1 - x_2)} = |x_1 - x_2|.$$

In this case the distance between every sequential pair of integers is the same,

$$\begin{aligned} d(x_{n+1}, x_n) &= |x_{n+1} - x_n| \\ &= |(x_n + 1) - x_n| \\ &= |1| \\ &= 1, \end{aligned}$$

endowing the space with a particular uniformity.

Because this metric enforces a minimum distance between every pair of elements it is compatible with the discrete topology. In particular the atomic sets in the discrete topology maintain a strong separation between even neighboring elements.

This characterization of the integers is all well and good until we consider permuting the elements. Sorting the elements differently changes the ordering and algebraic operations, and the modified algebraic structure can define a different metric structure. In general only the discrete topology survives.

Despite all of this chaos, however, the behavior of this new space still conforms with our intuitions about the integers. In other words we've reached a bit of an existential ambiguity. Every countably infinite set equipped with the discrete topology and *any* choice of compatible strict ordering, arithmetic operations, and metric defines a space which exhibits all of the features we expect of the integers!

This leaves us with two possible interpretations of the “integers”. On one hand we can interpret the integers not as a single space but rather a *class* of spaces, with each choice of compatible non-topological structure defining a *distinct* instance. From this perspective we can't speak about *the* integers so much as *which* integers we use in any given application.

Alternatively we can think of the integers as a single *flexible* space with the different choices of non-topological structure defining different *configurations*. In this case it does make sense

to talk about *the* integers, but at the same that doesn't specify a configuration and hence how the integers should be used in practice.

Regardless of which interpretation we embrace, “the” integers admit a few particularly useful subspaces. In particular we can construct the **natural numbers**,

$$\mathbb{N} = \{x \in \mathbb{Z} \mid x \geq 0\},$$

by removing the elements smaller than zero, and the **strictly positive natural numbers**

$$\mathbb{N}^+ = \{x \in \mathbb{Z} \mid x > 0\}$$

by removing the zero element as well. These subspaces inherit ordering, metric, and topological structure from the integers but the algebraic structure will be complicated a bit. For example we can't consistently define subtraction on the natural numbers because the difference between two positive integers can be negative, which falls outside of \mathbb{N} .

2.3 Real Lines

The prototypical model for a continuum that appears to contain endless points no matter how closely we examine it is the **real line**, \mathbb{R} , which is also known as the **real numbers** and a **one-dimensional Euclidean space**. This space sets the stage for important constructions like differential and integral calculus, but it also exhibits some subtle properties that can compromise our intuitions and require a bit of careful technical detail at times to avoid inconsistent behavior.

Formally the real line is built from an uncountably infinite set of elements that, like the integers, is equipped with a strict ordering, algebra, metric, and topology. The algebraic structure includes addition and multiplication operators which can also be used to define corresponding subtraction and division operators.

At the same time the binary multiplication operator also satisfies the criteria for a scalar multiplication operator, making the real line a vector space as well. From a linear algebraic perspective we can interpret a real line not as a collection of points but rather a collection of arrows stretching from the zero to each point.

As with the integers the algebraic structure on the real line can be used to construct a metric,

$$d(x_1, x_2) = \sqrt{(x_1 - x_2) \cdot (x_1 - x_2)},$$

which then defines a compatible metric topology.

Unfortunately this construction is not unique, and the real line suffers from the same interpretational ambiguity as the integers. Every uncountably infinite set equipped with a certain non-discrete, Hausdorff topology and *any* choice of compatible ordering, algebra, and metric

defines a space that behaves like a real line. Consequently we have to accept that \mathbb{R} does not completely specify a space.

On one hand we can take \mathbb{R} to denote a class of real lines, with infinitely many distinct realizations defined by different choices of ordering, algebra, and metric. On the other hand we can interpret \mathbb{R} as a single, flexible space that can be configured in an infinite number of different ways by choosing different auxiliary structures. In this latter perspective the different configurations are also known as different **parameterizations** of the real line.

The power set of a real line contains a wealth of subsets that we can use to construct useful subspaces. For example the integers are a subset of the real line, $\mathbb{Z} \subset \mathbb{R}$. We can also define the **positive real line** as

$$\mathbb{R}^+ = \{x \in \mathbb{R} \mid x \geq 0\}.$$

More generally we can define **intervals** that contain all points within an upper and lower bound. **Open intervals** do not contain the bounds,

$$(x_1, x_2) = \{x \in \mathbb{R} \mid x_1 < x < x_2\}$$

while **closed intervals** do,

$$[x_1, x_2] = \{x \in \mathbb{R} \mid x_1 \leq x \leq x_2\}.$$

All of these subspaces inherit topological, ordering, and metric structure from the initial real line, but they may not inherit all of the algebraic structure.

Regardless of the interpretation we can use the metric structure of a real line to construct a useful visualization of the space. In particular any partition of the real line into equally long intervals,

$$\dots, x_{-i}, \dots, x_{-1}, x_0, x_1, \dots, x_i, \dots$$

with

$$d(x_{i+1}, x_i) = \delta,$$

defines a **grid** that can be used to communicate its basic structure (Figure 17).

2.4 Extended Real Lines

One limitation of a real line is that it does not contains points that *approach* either negative or positive infinity, but not points that represent those limits directly. An **extended real line** resolves introduces two new elements, one smaller than all elements in the initial real line to represent negative infinity and one larger than all of the elements to represent positive infinity. In other words we can interpret an extended real line as the closed interval $[-\infty, \infty]$ while a real line is the corresponding open interval $(-\infty, \infty)$.

The inclusion of these two points does introduce some slight technical challenges. For example the algebraic structure is complicated by the fact that neither $-\infty$ nor $+\infty$ have well-defined

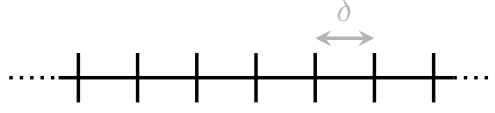


Figure 17: A grid of equally distant elements in a real line partitions the space into equally long intervals that communicates the metric structure.

additive or multiplicative inverses, unlike most other points. To avoid any inconsistencies we have to slightly tweak any structure derived from these operations, such as the metric and the corresponding metric topology.

Appending positive infinite to a positive real line defines an **extended positive real line** $[0, +\infty]$. We have already been sneakily introduced to the extended positive real line in [Chapter One](#). There we used it to model the potentially infinite reservoir that we want to allocate across a finite set. In the next chapter we will use it model a potentially infinite reservoir to be allocated across any set.

3 Relating Spaces

One we have defined spaces a natural question is how we can relate different spaces to each other. This requires relating not only the elements in the set but also any structure we have endowed onto that set.

3.1 Relating Sets

A **function**, also know as a **map** or **transformation**, is a relation between the elements of some *input* set and the elements of some *output* set (Figure 18). Mathematically we specify functions as $f : X \rightarrow Y$ where X denotes the input set and Y denotes the output set. If we want to be more detailed then we can also write

$$f : X \rightarrow Y$$

$$x \mapsto y = f(x),$$

where $x \in X$ is a variable taking values in the input set and $y \in Y$ is a variable taking values in the output set. This allows us to, for example, detail how exactly the function transforms x into y .

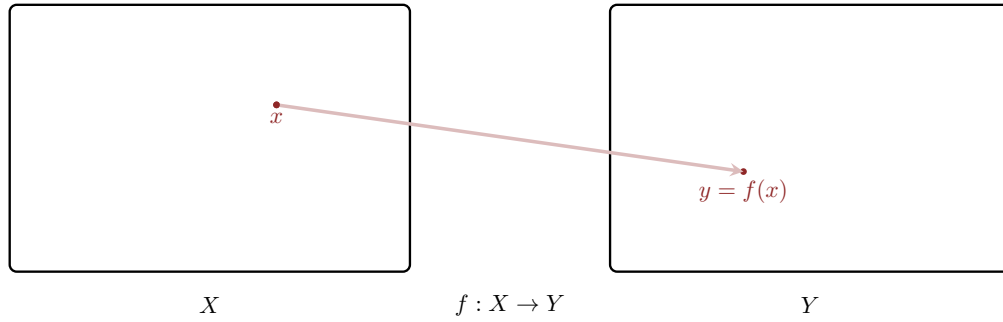


Figure 18: A function relates elements of an input set X to elements of an output set Y .

One potential source of confusion when working with functions is mistaking a function $f : X \rightarrow Y$ with the *evaluation* of a function on a given input variable, $y = f(\tilde{x}) \in Y$.

3.1.1 Classifying Functions

There are many ways to relate the elements from a given input set to the elements of a given output set. Some relations, however, are more useful than others. In order to understand the potential utility of different functions it helps to categorize them based on some basic properties.

For example if each input element maps to one, and only one, output element then equal function outputs will always imply equal function inputs and vice versa. More formally when $f(x_1) = f(x_2)$ if and only if $x_1 = x_2$ we say that the function f is **injective** (Figure 19). Non-injective functions are also known as **many-to-one** functions.

Similarly if each element of Y is the output of evaluating f on some input element in X then the function is said to be **surjective** (Figure 20).

When a function is both injective and surjective then *every* output element is uniquely paired with a single input element. In other words the input set and the output set are in perfect correspondence. These functions are known as **bijective** functions.

To demonstrate these behaviors let's consider four different functions that map the elements in a real interval $X = [0, 1]$ into the elements of another real interval $Y = [0, 1]$.

The function (Figure 21a)

$$f_1 : [0, 1] \rightarrow [0, 1]$$

$$x \mapsto \frac{1}{1 + 4 \cdot x \cdot (1 - x)}$$

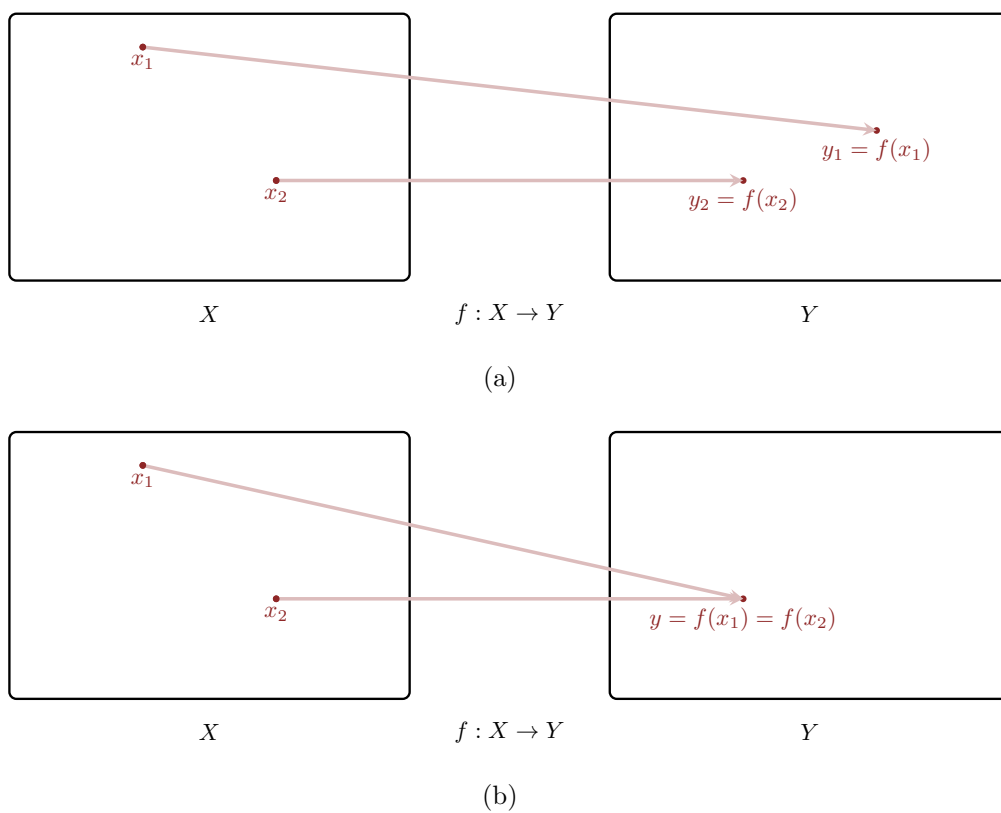


Figure 19: An (a) injective function maps each distinct input element to a distinct output element. A (b) non-injective function maps two or more input elements to the same output element.

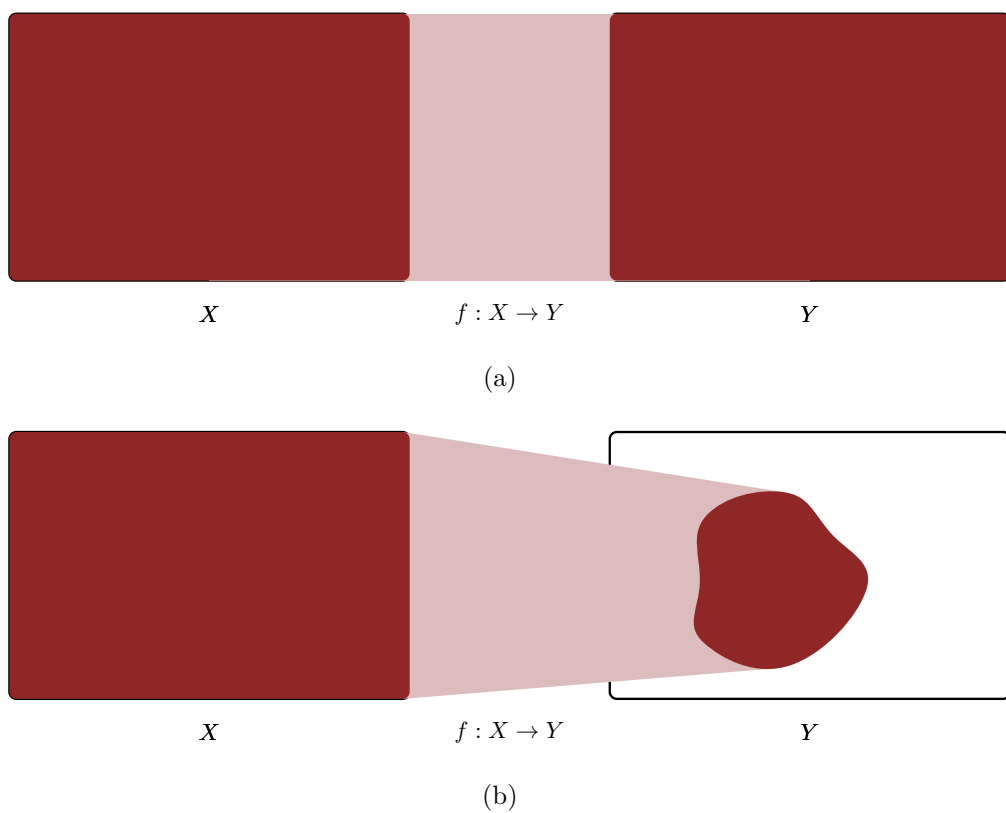
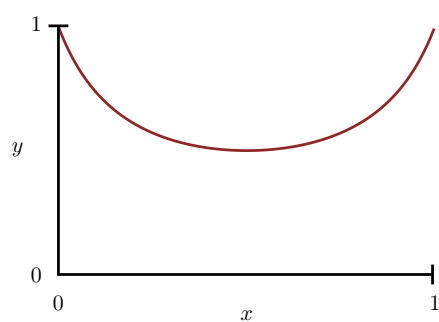
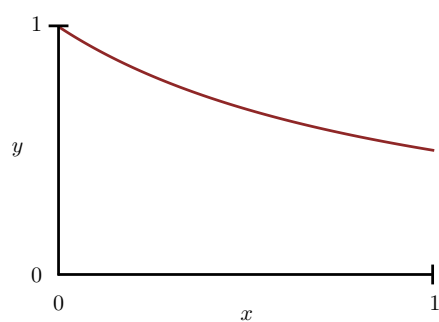


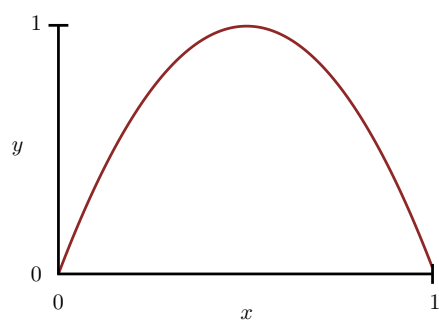
Figure 20: A (a) surjective function maps input elements to *every* element in the output set. A (b) non-surjective function maps input elements to only a subset of the output set.



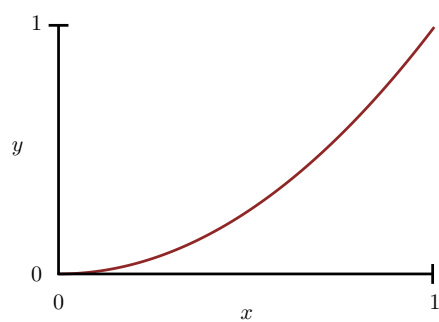
(a)



(b)



(c)



(d)

Figure 21: Injective functions map distinct input elements to distinct output elements while surjective functions map into every output element at least once. Here (a) is neither injective nor surjective, (b) is injective but not surjective, (c) is not injective but is surjective, and (d) is both injective and surjective.

is not injective because both x and $1 - x$ map to the same output element. Moreover it is not surjective because no input element will ever yield an output less than $y = \frac{1}{2}$.

Like f_1 the function (Figure 21b)

$$\begin{aligned} f_2 : [0, 1] &\rightarrow [0, 1] \\ x &\mapsto \frac{1}{1+x} \end{aligned}$$

outputs only values larger than or equal to $y = \frac{1}{2}$ and hence is not surjective. On the other hand each distinct input element maps to a distinct output element so unlike the function f_1 the function f_2 is injective.

This function (Figure 21c)

$$\begin{aligned} f_3 : [0, 1] &\rightarrow [0, 1] \\ x &\mapsto x \cdot (1 - x) \end{aligned}$$

shares the same symmetry as f_1 – because both x and $1 - x$ map to the same output element neither function is injective. Unlike f_1 the output elements of f_3 completely fill up the output set, making f_3 a surjective function.

Finally the function (Figure 21d)

$$\begin{aligned} f_4 : [0, 1] &\rightarrow [0, 1] \\ x &\mapsto x \cdot x \end{aligned}$$

is both injective *and* surjective, and hence bijective.

Functions that map the input set into itself, $f : X \rightarrow X$, are known as **automorphisms**. In other words a bijective automorphism doesn't change X but may permute the elements around. This permutation can in turn warp any structure endowed on X .

The **identity function** is the unique bijective automorphism that maps every element $x \in X$ back to itself,

$$\begin{aligned} \text{Id} : X &\rightarrow X \\ x &\mapsto x , \end{aligned}$$

leaving the set completely unchanged. When working with algebraic spaces we have to be careful to distinguish between the identity *function* that maps an input set into itself and the identity *element* of that set.

3.1.2 Function Inverses

Bijjective functions are particularly special because we can always undo their action and recover the input set. Formally if $f : X \rightarrow Y$ is a bijection then we can define an **inverse function** $f^{-1} : Y \rightarrow X$ such that

$$f^{-1}(f(x)) = x$$

for all $x \in X$. In other words no information is lost when we map from X to Y allowing us to completely recover X again if needed (Figure 22).

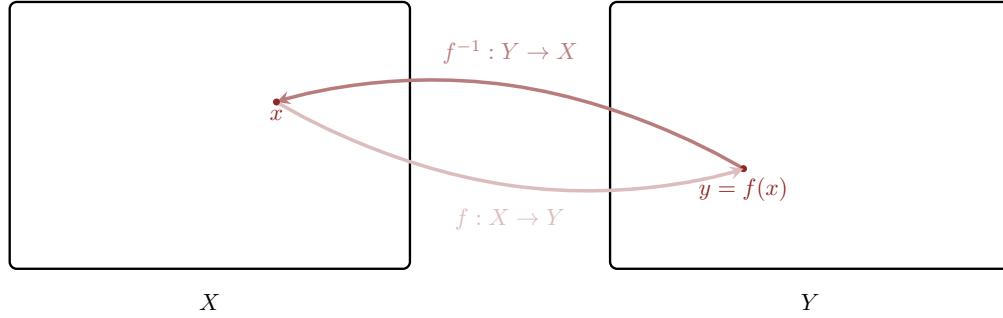


Figure 22: A bijective function f identifies every output element with one and only input element. Consequently we can construct an inverse function that takes each output element back to the corresponding input element.

As with the term “identity” we have to be careful when using the term “inverse” in the context of algebraic spaces. In particular we have to take care to distinguish between inverse *functions* between two sets and inverse *elements* in either of those two sets.

Without bijectivity we cannot define inverse functions that map distinct output elements to distinct input elements. If a function is not surjective then some output elements might not be related to *any* input elements. On the other hand if a function is not injective then some output elements may be related to *multiple* input elements.

Even if a function is not bijective, however, we can still define a generalized notion of inversion. Although we cannot generally relate elements $y \in Y$ to unique elements of X we can relate them to *subsets* of X that contain all of the elements that map into y ,

$$\begin{aligned} f^{-1} : Y &\rightarrow 2^X \\ y &\mapsto f^{-1}(y). \end{aligned}$$

The subset $f^{-1}(y) \subset X$ corresponding to a particular output element $y \in Y$ is referred to as a *preimage* or *fiber* of y (Figure 23). Intuitively we can reconstruct the entire input set by “weaving” these fibers together,

$$X = \cup_{y \in Y} f^{-1}(y).$$

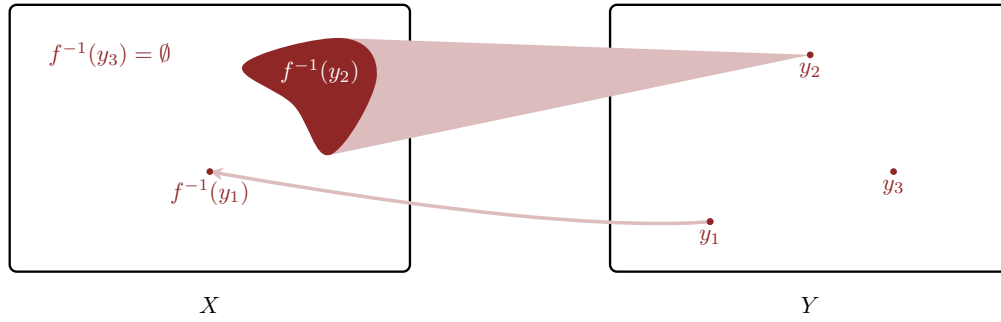


Figure 23: Even if a function is not bijective we can define a generalized inverse that maps each output element $y \in Y$ to a fiber of input elements $f^{-1}(y) \subset X$ that all map into y . In general these fibers can be empty, consist of a single input element, or consist of multiple input elements.

When f is not surjective then some fibers may be empty, but if f is surjective then all fibers will be non-empty. At the same time if a function is not injective then fibers will generally contain multiple elements, but if it is injective then all of the fibers will contain either one element or no elements. The fibers of bijective functions contain one and only one element: the result of evaluating the inverse function.

3.1.3 Function Composition

When defining a function inverse we had to take the output of a function $y = f(x)$ and used it as input to a second function, $f^{-1}(y) = f^{-1}(f(x))$. This chaining of functions with compatible input and output sets together can be applied much more generally and is known as **function composition** or just **composition** for short.

More formally given the functions $f : X \rightarrow Y$ and $g : Y \rightarrow Z$ we can construct the composition of f with g as the function

$$g \circ f : X \rightarrow Z$$

$$x \mapsto z = g(f(x)).$$

Using this notation we can define an inverse function a bit more compactly as

$$\text{Id} = f^{-1} \circ f.$$

In words the composition of a bijective function with its inverse function is the identify function.

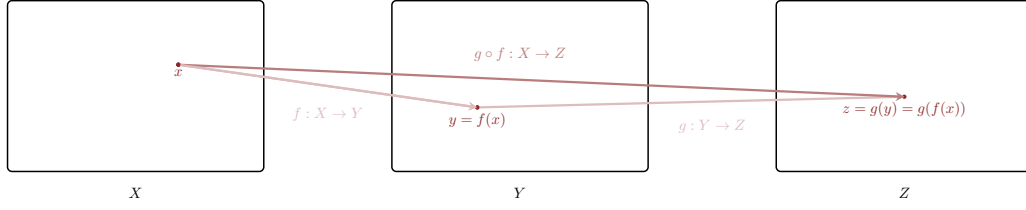


Figure 24: The functions $f: X \rightarrow Y$ and $g: Y \rightarrow Z$ can be chained together to map an input element $x \in X$ to an output element $z \in Z$. This combined mapping defines the composite function $g \circ f$.

3.2 Relating Structures

Functions define how the elements of an input set transform into the elements of an output set. If we want to relate entire *spaces*, however, then we also need to specify how the additional structures that define those spaces transform along with the underlying elements.

To be more precise let \mathfrak{x} denote whatever structure we endow on the set X so that our input space is fully described as (X, \mathfrak{x}) . In order to fully define an output space we need to map input elements in X to output elements in Y *and* equip that output space with some structure \mathfrak{y} . We can choose that structure ourselves, or we can try to derive it from the input structure \mathfrak{x} .

Fortunately maps between sets often *induce* natural maps between structures, allowing us to fully transform spaces without having to make additional assumptions about how structures transform. Unfortunately these induced maps don't always give the structure we might expect.

3.2.1 Pushforward and Pullback Relations

Consider a function $f: X \rightarrow Y$ that maps input elements into output elements and a subset of the input space $\mathfrak{x} \subset X$. Applying f to each element of \mathfrak{x} defines a subset of the output space $\mathfrak{y} \subset Y$,

$$\mathfrak{y} = \{y \in Y \mid y = f(x) \text{ for some } x \in \mathfrak{x}\}$$

In other words the element-wise evaluation of f implicitly defines a map from input subsets to output subsets (Figure 25a)

$$\begin{aligned} f_*: 2^X &\rightarrow 2^Y \\ \mathfrak{x} &\mapsto \mathfrak{y} = f_*(\mathfrak{x}). \end{aligned}$$

If f is not bijective then we cannot map output elements to unique input elements. We can, however, always map every output element to a, possibly empty, collection of input elements

$f^{-1}(y) \subset X$. The union of the fibers from all elements in an output subset $y \subset Y$ defines a subset of all of the input elements that map into y ,

$$x = \cup_{y \in y} f^{-1}(y) \subset X,$$

or equivalently

$$x = \{x \in X \mid f(x) \in y\}.$$

Weaving together these fibers implicitly defines a map from output subsets to input subsets (Figure 25b)

$$\begin{aligned} f^* : 2^Y &\rightarrow 2^X \\ y &\mapsto x = f^*(y). \end{aligned}$$

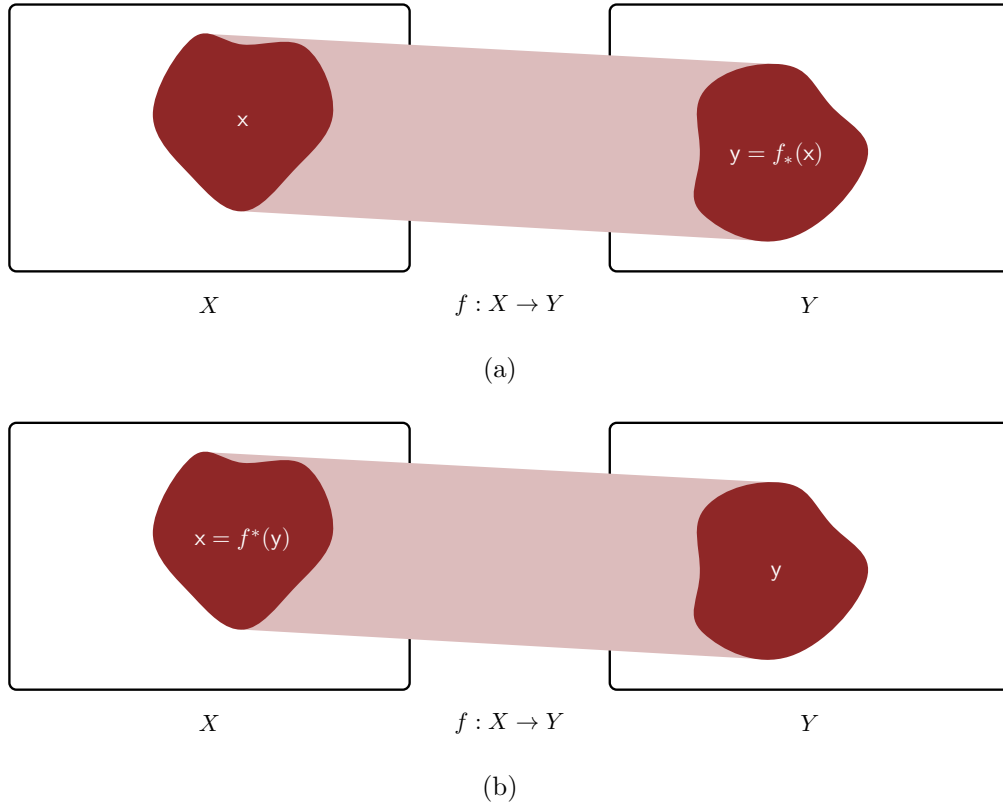


Figure 25: The action of a function f on individual elements induces an action on entire subsets.

In particular we can construct (a) a map from input subsets to output subsets and (b) a map from output subsets to input subsets.

Conveniently this pattern is not exceptional. Sufficiently well-behaved functions that map an input set into an output set often induce mappings from objects defined over that input set to

objects defined over that output set or vice versa. This includes not only subsets but also all of the structures that elevate a set into a space that we introduced in [Section 1.2](#).

To be a bit more formal let X be the set defining an input space and \mathfrak{x} the particular structure, or structures, that elevates that set to a space so that we can completely specify our space as (X, \mathfrak{x}) . Moreover denote the set of all structures that we can endow onto the input space X as \mathfrak{X} and the set of all structures that we can endow onto the output space Y as \mathfrak{Y} .

The subset map $f_* : 2^X \rightarrow 2^Y$ that we constructed above *pushes* inputs subsets *forward* along the action of the function f . Analogously any relation between input structures and output structures induced by $f : X \rightarrow Y$ that pushes forward input structures into compatible output structures,

$$\begin{aligned} f_* : \mathfrak{X} &\rightarrow \mathfrak{Y} \\ \mathfrak{x} &\mapsto \mathfrak{y} , \end{aligned}$$

is denoted a **pushforward function along f** . When the underlying function f is unambiguous then we also sometimes refer to f_* as a **pushforward function**.

As with functions that act on sets we have to be careful to distinguish between pushforward *functions* $f_* : \mathfrak{X} \rightarrow \mathfrak{Y}$ and the particular pushforward *structure* defined by evaluating the pushforward function on a given input structure, $f_*(\mathfrak{x})$. In other words if there might be any ambiguity then we have to be careful to use “pushforward” as an adjective, explicitly modifying “function” or “structure”, and not an ambiguous noun.

Similarly the subset map $f^* : 2^Y \rightarrow 2^X$ that we constructed above *pulls* output subsets *back* against the action of the function f . More generally any relation between output structures and input structures induced by $f : X \rightarrow Y$ that pulls output structures back into compatible input structures,

$$\begin{aligned} f^* : \mathfrak{Y} &\rightarrow \mathfrak{X} \\ \mathfrak{y} &\mapsto \mathfrak{x} , \end{aligned}$$

is denoted a **pullback function against f** . When the underlying function f is unambiguous then we also sometimes refer to f^* as a **pullback function**.

The existence of pushforward and pullback functions depends on how the defining function $f : X \rightarrow Y$ interacts with the relevant structure. In general the nicer the function f the more structure transforming functions it induces.

3.2.1.1 Relating Orderings

Consider, for example, orderings. When f is an injective function every input element x corresponds to one and only one output element, $y = f(x)$. Consequently any comparison

between output elements can be translated to the corresponding input elements. If the output space Y is equipped with an ordering $<$ then we can define a compatible ordering on X as

$$x_1 \prec^* x_2$$

if and only if

$$f(x_1) < f(x_2).$$

In other words every injective function allows us to pullback orderings on the output set $<_Y$ to orderings on the input set $<_X = f^* <_Y$.

Unless f is also surjective, and hence bijective, we cannot pushforward orderings on the input space to orderings on the output space. The problem is that if any pair of output elements are associated with multiple input elements,

$$\begin{aligned} y_1 &= f(x_1) = f(x'_1) \\ y_2 &= f(x_2) = f(x'_2), \end{aligned}$$

then we can have conflicting input orderings,

$$\begin{aligned} x_1 &< x_2 \\ x'_2 &< x'_1, \end{aligned}$$

that cannot be combined into a consistent output ordering.

Ultimately injective functions define ordering pullbacks but only bijective functions define ordering pushforwards and ordering pullbacks.

3.2.1.2 Relating Algebras

Algebraic structure is a bit more difficult to translate. Consider, for example, a binary algebraic operation that takes two elements from a set and returns a single element. In order to pushforward an operation on the input space \cdot we would need to map output elements to input elements, apply the operation, and then map the result back to the output space,

$$y_1 (f_* \cdot) y_2 \equiv f(f^{-1}(y_1) \cdot f^{-1}(y_2)).$$

If f is not bijective, however, then the inverses will be ill-defined.

Similarly in order to pullback an operation on the output space \cdot we would need to map input elements to output elements, apply the operation, and then map the result back to the input space,

$$x_1 (f^* \cdot) x_2 \equiv f^{-1}(f(x_1) \cdot f(x_2)).$$

An injective function is sufficient for the first two steps to be well-defined, but we still need f to be bijective in order for the last step to also be well-defined.

In other words we can pushforward and pullback binary algebraic operations only when f is a bijection.

3.2.1.3 Relating Metrics

Metrics are a bit more forgiving. Applying $f : X \rightarrow Y$ to two input elements defines two output elements that we can plug into a metric defined over the output set,

$$(f^*d)(x_1, x_2) \equiv d(f(x_1), f(x_2)).$$

This composition, however, does not always define a proper metric over the input set! In particular if f is not injective then $d(f(x_1), f(x_2))$ can vanish even if $x_1 \neq x_2$. Fortunately if f is injective then $(f^*d)(x_1, x_2)$ will satisfy all of the properties of a metric on the input set in which case we denote it the **pullback metric**.

3.2.1.4 Relating Topologies

Finally let's consider how we might translate topologies across a function. Using the map from input subsets to output subsets that we constructed above we can map the collection of subsets that define a topology on the input set,

$$\mathbf{t} = \{x_1, \dots, x_i, \dots\},$$

into a collection of subsets in the output set,

$$\{f^*(x_1), \dots, f^*(x_i), \dots\}.$$

This output collection, however, will not always satisfy the properties of a topology. For example if f is not surjective then the output collection will not contain the full set as needed. Even if f is surjective the pushforward sets are not guaranteed to be closed under unions and intersections and hence will not, in general, define a topology.

Fortunately pullbacks sets are a bit more well-behaved. Pulling back the subsets that define a topology on the output set,

$$\mathbf{t} = \{y_1, \dots, y_i, \dots\},$$

gives a collection of subsets on the input set,

$$\{f_*(y_1), \dots, f_*(y_i), \dots\}.$$

When f is surjective then this collection will include both $f^*(\emptyset) = \emptyset$ and $f^*(Y) = X$. Moreover these pullback subsets are always closed under countable unions and finite intersections so long as the input subsets are. In other words if f is surjective then

$$f^*\mathbf{t} \equiv \{f_*(y_1), \dots, f_*(y_i), \dots\}$$

defines a **pullback topology** on X .

3.2.2 Bijective Pushforwards and Pullbacks Relations

The asymmetry between the availability of pushforward and pullback actions for different structures can be frustrating in practice, especially when an application supplies structure on the wrong set. Fortunately this asymmetry washes away when we consider bijective functions and their inverses.

Consider for example some structure that only pulls back against a function,

$$f^* : \mathfrak{Y} \rightarrow \mathfrak{X}.$$

If f is bijective then we can pullback that structure against the inverse function,

$$(f^{-1})^* : \mathfrak{X} \rightarrow \mathfrak{Y},$$

which implicitly pushes forward structure defined on X to structure defined on Y .

Similarly if we have some structure that only pushes forward along a function,

$$f_* : \mathfrak{X} \rightarrow \mathfrak{Y},$$

then we can use the pushforward along the inverse function

$$(f^{-1})_* : \mathfrak{Y} \rightarrow \mathfrak{X},$$

to pullback structure defined on Y to structure defined on X .

Importantly these induces transformations are also inverses,

$$\begin{aligned} f^* \circ (f^{-1})^* &= \text{Id}_{\mathfrak{X}} \\ (f^{-1})^* \circ f^* &= \text{Id}_{\mathfrak{Y}} \\ (f^{-1})_* \circ f_* &= \text{Id}_{\mathfrak{X}} \\ f_* \circ (f^{-1})_* &= \text{Id}_{\mathfrak{Y}}. \end{aligned}$$

In other words bijective functions allow us to move back and forth between the input space and the output space without losing information about the individual element or any structure endowed to those elements.

3.2.3 Structure-Preserving Relations

Pushforward and pullback functions allow us to *lift* a transformation between sets into a transformation between spaces. For structure that can be pushed forward along the function $f : X \rightarrow Y$ any input space (X, \mathfrak{x}) automatically defines a compatible output space $(Y, f_*(\mathfrak{x}))$. Similarly for structure that can be pulled back against f any output space (Y, \mathfrak{y}) automatically defines a compatible input space $(X, f^*(\mathfrak{y}))$.

Many applications, however, don't just define only an input space (X, \mathfrak{x}) or only an output space (Y, \mathfrak{y}) but rather *both*. In this case pushforward structure may not be compatible with the structure that has already been endowed on the output space,

$$f_*(\mathfrak{x}) \neq \mathfrak{y},$$

and pullback structure may not be compatible with the structure that has already been endowed on the input space,

$$f^*(\mathfrak{y}) \neq \mathfrak{x}.$$

The exceptional functions that yield compatible pushforward and/or pullback structure are known as **structure-preserving transformations**. Structure-preserving bijections in particular are also known as **isomorphisms**.

3.2.3.1 Ordering-Preserving Relations

For example consider an ordered input space $(X, <_X)$ and an ordered output space $(Y, <_Y)$. For any function $f : X \rightarrow Y$ the pullback ordering $f^*<_Y$ is defined by

$$x_1 \ f^*<_Y \ x_2$$

if and only if

$$f(x_1) <_Y f(x_2).$$

There's no guarantee, however, that this will be consistent with the pre-specified ordering $<_X$. Instead we will have $f_*<_Y = <_X$ only for the exceptional functions satisfying

$$f(x_1) <_Y f(x_2)$$

if and only if

$$x_1 <_X x_2.$$

These functions are known as **monotonically increasing functions**. Functions that respect given input and output partial orderings are known as **monotonically non-decreasing functions**.

To demonstrate consider the functions

$$\begin{aligned} f_1 : \mathbb{R} &\rightarrow \mathbb{R} \\ x &\mapsto x \cdot x \cdot x \end{aligned}$$

and

$$\begin{aligned} f_2 : \mathbb{R} &\rightarrow \mathbb{R} \\ x &\mapsto -x \cdot x \cdot x \end{aligned}$$

that map between two distinct real lines with distinct orderings. When $x_1 <_X x_2$ we have $f_1(x_1) <_Y f_1(x_2)$ but $f_2(x_2) <_Y f_2(x_1)$ making the former monotonically increasing but the latter not (Figure 26).

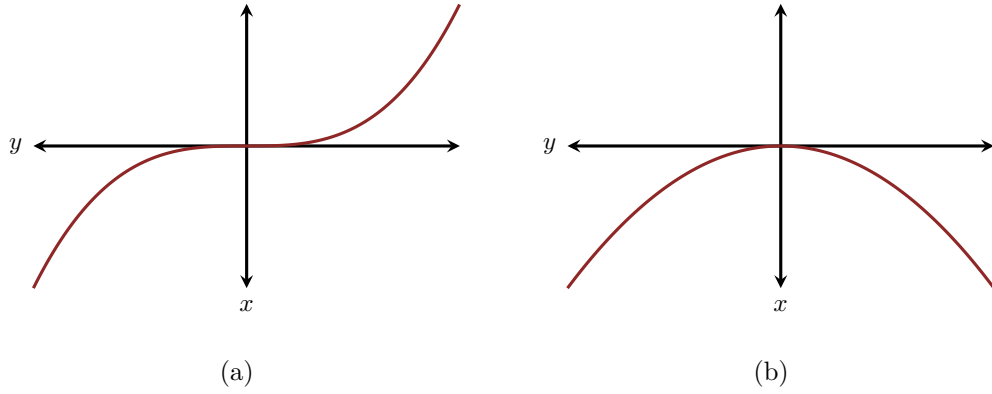


Figure 26: Monotonically increasing functions preserve orderings so that larger inputs always imply larger outputs. The function (a) $f_1 : x \mapsto x^3$ is monotonic but the function (b) $f_1 : x \mapsto -x^3$ is not.

3.2.3.2 Algebra-Preserving Relations

A bijective function that preserves algebraic structure is known as a **homomorphism**. Given two algebraic spaces equipped with binary algebraic operations, (X, \cdot_X) and (Y, \cdot_Y) , a function $f : X \rightarrow Y$ is homomorphic if and only if

$$f(x_1 \cdot_X x_2) = f(x_1) \cdot_Y f(x_2).$$

Consider for example the cubic function that maps a real line into another real line,

$$\begin{aligned} f : \mathbb{R} &\rightarrow \mathbb{R} \\ x &\mapsto x \cdot x \cdot x. \end{aligned}$$

Because

$$(x_1 + x_2)^3 \neq x_1^3 + x_2^3$$

, where the $+$ on the left denotes addition on the input space and the $+$ on the right denotes addition on the output space, f is not a homomorphism for the addition operation. On the other hand because

$$(x_1 \cdot x_2)^3 = x_1^3 \cdot x_2^3,$$

where again \cdot on the left and right actually denotes two distinct multiplication operations, the function f is a homomorphism between for the multiplication operation.

3.2.3.3 Metric-Preserving Relations

Injective transformations that preserve metric structure, and hence distances between input and output elements

$$d_X(x_1, x_2) = d_Y(f(x_1), f(x_2)),$$

are known as **isometries**. Non-isometric functions appear to *warp* distances relative to metric defined on the output space, and hence any grid that we might use to visualize the input space.

For example the linear function

$$\begin{aligned} f_1 : \mathbb{R} &\rightarrow \mathbb{R} \\ x &\mapsto x - \frac{1}{4} \end{aligned}$$

is an isometry for the natural metric over the real line,

$$\begin{aligned} d_Y(f(x_1), f(x_2)) &= \sqrt{(f(x_1) - f(x_2)) \cdot (f(x_1) - f(x_2))} \\ &= \sqrt{\left((x_1 - \frac{1}{4}) - (x_2 - \frac{1}{4})\right) \cdot \left((x_1 - \frac{1}{4}) - (x_2 - \frac{1}{4})\right)} \\ &= \sqrt{(x_1 - x_2) \cdot (x_1 - x_2)} \\ &= d_X(x_1, x_2). \end{aligned}$$

One consequence of this isometry is that any grid of equally distant input elements will map into a grid of equally distant output elements (Figure 27a).

On the other hand the bijective cubic function

$$\begin{aligned} f_2 : \mathbb{R} &\rightarrow \mathbb{R} \\ x &\mapsto x \cdot x \cdot x \end{aligned}$$

is not an isometry. Instead the pushforward of the input metric defines different distances than the output metric, resulting in a warped grids (Figure 27b).

3.2.3.4 Topology-Preserving Relations

Bijections that respect any topological structure endowed to the input and output sets are denoted **homeomorphisms** which, unfortunately, we have to avoid confusing with algebra-preserving *homomorphisms*. Etymologically “homomorphism” is derived from the Greek for “same shape” while “homeomorphism” is derived from the Greek for “similar shape” which have almost equivalent meanings. Contemporary mathematics, however, has adopted a convention where the two terms are used exclusively to refer to algebraic and topological structure, respectively.

More formally a bijective function is a homeomorphism if every subset in the input topology pushes forward to a subset in the output topology and vice versa. In other words neighborhoods

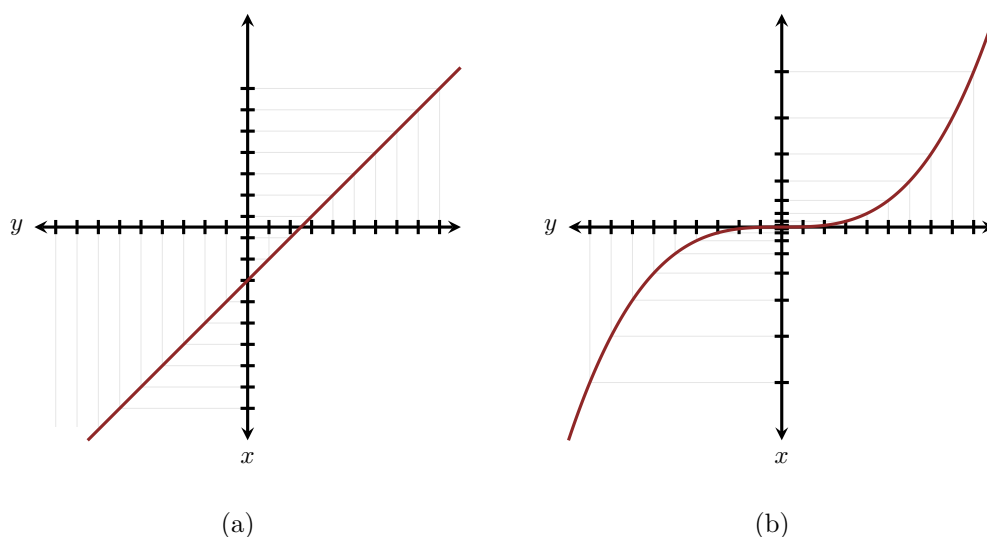


Figure 27: Isometries preserve the metric structure of the input and output spaces, ensuring that equally spaced input elements map into equally spaced output elements. When the input and output space are real lines (a) linear functions define isometries and where as (b) nonlinear functions warp the input metric relative to the output metric.

are preserved and input elements near each other are mapped into output elements that are also near each other, preserving the continuity of the space. Because of this homeomorphisms are also known as invertible **continuous** functions.

In [Section 1.2.4.3](#) we saw that lines and circles can be constructed from a common set equipped with different topologies. We can construct bijections between the underlying set elements but any homeomorphism will preserve the shape induced by the initial topology, mapping lines into lines or circles into circles. Only non-homeomorphic bijections are capable of transforming the defining topologies, and hence a line into a circle or vice versa ([Figure 28](#)).

3.2.4 Reparameterizations

The ambiguity in how we can interpret the integers and the real line as spaces that we encountered in [Section 2](#) also implies an ambiguity in how we interpret transformation between them.

Considering for example if we take the perspective where there is not a single real line but rather a collection of distinct real lines, each with their own defining ordering, algebra, and metric. In general a homeomorphism transforms one of these real lines into another, and only the exceptional monotonically-increasing, homomorphic, and isometric homeomorphisms will map one of these real lines into itself.

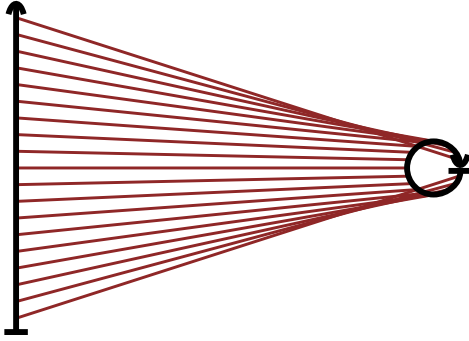


Figure 28: A line and a circle can be build from a common set but require distinct topologies to give them their characteristic shapes. Because homeomorphisms cannot modify these topologies they can map only lines into lines and circles into circles but not lines into circles. Any bijection from the elements that comprise a line and the elements that comprise a circle cannot be homeomorphic.

Conversely we can treat the real line as a flexible space defined by a fixed set and topology, with the different orderings, algebras, and metrics defining different configurations of the real line. From this perspective homeomorphisms maintain the defining structure but generally transform the auxiliary structures. This maps one configuration into another, effectively *re-configuring* the flexible real line. Because configurations are also known as parameterizations the homeomorphic transformations are also known as **reparameterizations**.

While this interpretational ambiguity is often taken for granted it does manifest in a variety of different applications. For example in physics we can interpret transformations of a physical system as *active* or *passive*. An active transformation is interpreted as changing the physical system while a passive transformation is interpreted as changing how we represent a fixed physical system. Both interpretations yield equivalent results, but one can be more natural in some contexts than others.

A variable taking values on the real line can be used to quantify some property of some physical system. Transforming from one real line to another can be interpreted as changing the actual quantification or equivalently as a change of *units* for a fixed quantification.

At the same time we find can find similar ambiguities in language. We might interpret text written in a particular language as fundamental in which case *translations* from one language to another map one text to another. That said we might also interpret these texts as *representations* of some universal statement in which case translations don't change the statement but do change the representation.

Similarly we might have the freedom to interpret computer programs written in different programming languages as either distinct programs or different implementations of some fixed

program with universal inputs and outputs. In the former interpretation *transpilers* map from one program to another but in the latter they map from one implementation to another.

Again these interpretations are equivalent, and we are free to adopt the one that is most useful in any given application. In this book I will embrace the perspective where the real line is defined as a flexible space, with different choices of ordering, algebraic, and metric structure defining different parameterizations of that space. This will have some subtle consequences going forwards, but I will address those consequences as they arise.

4 Conclusion

Spaces are one of those ubiquitous mathematical concepts that are often taken for granted. While this can be adequate for more heuristic mathematical modeling tasks it does leave us open to some existential crises if we look a little bit too closely and start asking, for example, what exactly a reparameterization is.

In this chapter we've surveyed the formal construction of spaces, from an underlying set to the many structures that endow that set with useful properties, and their transformations. Along the way we took a careful look at familiar concepts like variables while also digging into some more complicated concepts like topology with an emphasis on appreciating how they can be useful beyond just theoretical mathematics. This understanding will provide a strong foundation on which we introduce probability theory in its full generality, and hence full power.

There are a wealth of resources for those interested in exploring these topics more deeply, although finding the right resource is often a challenge. Many of the more accessible resources are prone to oversimplification which can make it difficult to iterate to more technical resources. At the same time the more technical resources often jump directly into detail with little to no context for why the details might be relevant.

Personally I have found John Lee's introductory writing to be a reasonable middle ground, especially the appendices and first few chapters of Lee (2011) as well as the appendices to Lee (2013). Similarly I find that Curtis (1993) provides an accessible introduction to the more technical aspects of linear algebra. An additional benefit of these resources is that they can help acclimate one to the writing style particular to theoretical mathematics which can make other resources a bit more accessible.

At the same time the articles on [Wikipedia](#) are continuously improving. I find these articles to be a particularly useful reference for delving into technical details on precise topics.

Acknowledgements

I thank Simon Duane, Jerzy Baranowski, Ron Garcia, and Stone Chen for helpful comments and discussion.

A very special thanks to everyone supporting me on Patreon: Adam Fleischhacker, Adriano Yoshino, Alan Chang, Alessandro Varacca, Alexander Bartik, Alexander Noll, Alexander Petrov, Alexander Rosteck, Anders Valind, Andrea Serafino, Andrew Mascioli, Andrew Rouillard, Andrew Vigotsky, Angie_Hyunji Moon, Ara Winter, Austin Rochford, Austin Rochford, Avraham Adler, Ben Matthews, Ben Swallow, Benjamin Glemain, Bradley Kolb, Brynjolfur Gauti Jónsson, Cameron Smith, Canaan Breiss, Cat Shark, Charles Naylor, Chase Dwelle, Chris Zawora, Christopher Mehrvarzi, Chuck Carlson, Colin Carroll, Colin McAuliffe, Cruz, Damien Mannion, Damon Bayer, dan mackinlay, Dan Muck, Dan W Joyce, Dan Waxman, Dan Weitzenfeld, Daniel Edward Marthaler, Daniel Rowe, Darshan Pandit, Darthmaluus , David Burdelski, David Galley, David Humeau, David Wurtz, dilsher singh dhillon, Doug Rivers, Dr. Jobo, Dr. Omri Har Shemesh, Ed Cashin, Ed Henry, Edgar Merkle, edith darin, Eric LaMotte, Erik Banek, Ero Carrera, Eugene O’Friel, Felipe González, Fergus Chadwick, Finn Lindgren, Florian Wellmann, Francesco Corona, Geoff Rollins, Greg Sutcliffe, Guido Biele, Hamed Bastan-Hagh, Haonan Zhu, Hector Munoz, Henri Wallen, hs, Hugo Botha, Håkan Johansson, Ian Costley, Ian Koller, idontgetoutmuch, Ignacio Vera, Ilaria Prosdocimi, Isaac Vock, J, J Michael Burgess, Jair Andrade, James Hodgson, James McInerney, James Wade, Janek Berger, Jason Martin, Jason Pekos, Jason Wong, Jeff Burnett, Jeff Dotson, Jeff Helzner, Jeffrey Erlich, Jesse Wolfhagen, Jessica Graves, Joe Wagner, John Fournoy, Jonathan H. Morgan, Jonathon Vallejo, Joran Jongerling, Joseph Despres, Josh Weinstock, Joshua Duncan, Joshua Griffith, Josué Mendoza, JU, Justin Bois, Karim Naguib, Karim Osman, Ke-jia Shi, Kevin Foley, Kristian Gårdhus Wichmann, Kádár András, lizzie , LOU ODETTE, Marc Dotson, Marcel Lüthi, Marek Kwiatkowski, Mark Donoghoe, Mark Worrall, Markus P., Martin Modrák, Matt Moores, Matthew, Matthew Kay, Matthieu LEROY, Maurits van der Meer, Merlin Noel Heidemanns, Michael DeWitt, Michael Dillon, Michael Lerner, Mick Cooney, Márton Vaitkus, N Sanders, Name, Nathaniel Burbank, Nic Fishman, Nicholas Clark, Nicholas Cowie, Nick S, Nicolas Frisby, Octavio Medina, Ole Rogeberg, Oliver Crook, Olivier Ma, Pablo León Villagrà, Patrick Kelley, Patrick Boehnke, Pau Pereira Batlle, Peter Smits, Pieter van den Berg , ptr, Putra Manggala, Ramiro Barrantes Reynolds, Ravin Kumar, Raúl Peralta Lozada, Riccardo Fusaroli, Richard Nerland, RLW, Robert Frost, Robert Goldman, Robert kohn, Robert Mitchell V, Robin Taylor, Ross McCullough, Ryan Grossman, Rémi , S Hong, Scott Block, Scott Brown, Sean Pinkney, Sean Wilson, Seth Axen, shira, Simon Duane, Simon Lilburn, Srivatsa Srinath, sssz, Stan_user, Stefan, Stephanie Fitzgerald, Stephen Lienhard, Steve Bertolani, Stone Chen, Susan Holmes, Svilup, Sören Berg, Tao Ye, Tate Tunstall, Tatsuo Okubo, Teresa Ortiz, Thomas Lees, Thomas Vladeck, Tiago Cabaço, Tim Radtke, To-bychev , Tom McEwen, Tony Wuersch, Utku Turk, Virginia Fisher, Vitaly Druker, Vladimir Markov, Wil Yegelwel, Will Farr, Will Tudor-Evans, woejozney, Xianda Sun, yolhaj , yureq , Zach A, Zad Rafi, and Zhengchen Cai.

References

- Curtis, Charles W. 1993. *Linear Algebra*. Fourth. Undergraduate Texts in Mathematics. Springer-Verlag, New York.
- Lee, John M. 2011. *Introduction to Topological Manifolds*. Springer.
- . 2013. *Introduction to Smooth Manifolds*. Springer.

License

A repository containing all of the files used to generate this chapter is available on [GitHub](#).

The text and figures in this chapter are copyrighted by Michael Betancourt and licensed under the CC BY-NC 4.0 license:

<https://creativecommons.org/licenses/by-nc/4.0/>