# 1 Tensors and Their Subparts

Tensors are multiway arrays and serve as useful tools for data representation and analysis. Tensor decompositions are similar in spirit to matrix decompositions, such as principal component analysis (PCA), singular value decomposition (SVD), and nonnegative matrix factorization (NMF). If we consider that a matrix might generically represent objects (rows) and attributes (columns), the addition of multiple measurements at different times or in different scenarios can produce a multiway array that we refer to as a tensor. In 1952, Cattell proposed that data might be organized as

$$\boxed{\text{object}} \times \boxed{\text{feature}} \times \boxed{\text{scenario}}\,.$$

The tensor in this case might look like what we see in Fig. 1.1.

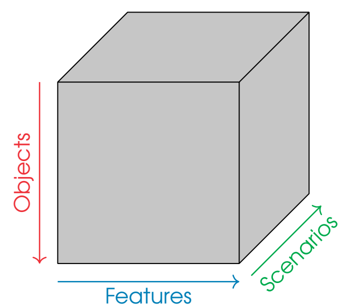


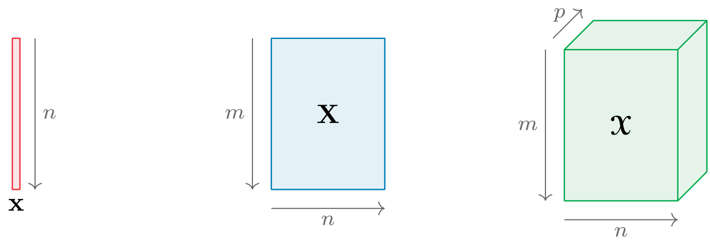

**Figure 1.1** Prototypical format of the tensor in data analysis.

The different scenarios might consist of measurements at different times or under different conditions. Furthermore, there is no reason to be constrained to organizing data into 3-way arrays.

The focus of this chapter is on understanding and manipulating tensor objects. A tensor is a multidimensional array, but it is oftentimes useful for considerations of storage or computation to view it in other ways, rearranging its entries as a vector or a matrix. We can potentially exploit structure such as sparsity or symmetry. Moreover, we can consider particular subparts of the tensor, called fibers, slices, and hyperslices. We describe several example tensors that we will revisit throughout the book. We close this chapter with a preview of the two main tensor decompositions discussed in this book: Tucker and CP.



 https://doi.org/10.1017/9781009471664.003




**(a)** Vector $\mathbf{x} \in \mathbb{R}^n$ is a tensor of order 1.

**(b)** Matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ is a tensor of order 2.

**(c)** Tensor $\mathcal{X} \in \mathbb{R}^{m \times n \times p}$ is of order 3.

**Figure 1.2** Tensors of order one, two, and three.

## 1.1 What Is a Tensor?

A **tensor** is a $d$-way array, where $d$ is referred to as the **order** of the tensor. Let's talk about how tensors relate to the known realm of vectors and matrices. First, a bit of notation. We denote the set of real values as $\mathbb{R}$. We represent scalars throughout as lowercase letters. We generally use the letters $i, j, k, \ell$ as indices into arrays and the letters $m, n, p, q, r, s$ to represent sizes. We assume that indices start from 1 (rather than 0). Additionally, we use the shorthand $[n] \equiv \{ 1, \ldots, n \}$, and we write $[m] \otimes [n] = \{ (i, j) \mid i \in [m], j \in [n] \}$.

**Definition 1.1** (Tensor) A **tensor** is a $d$-way array, and $d$ is the **order** of a tensor.

A **vector** is a one-dimensional array of numbers that represents a collection of measurements. In machine learning, a *feature vector* is the set of measurements that is used to characterize an object. We represent vectors throughout by lowercase boldface roman letters. If $\mathbf{x}$ is a real-valued vector of size $n$, then we write $\mathbf{x} \in \mathbb{R}^n$. Entry $i \in [n]$ of $\mathbf{x}$ is denoted as $\mathbf{x}(i)$ or compactly as $x_i$. A vector is a tensor of order 1.

A **matrix** is a two-dimensional array of numbers, such as a collection of feature vectors. We represent matrices throughout by uppercase boldface roman letters. If $\mathbf{X}$ is a real-valued matrix of size $m \times n$, then we write $\mathbf{X} \in \mathbb{R}^{m \times n}$. For instance, given a set of $m$ objects, each of which has $n$ features, the matrix entry $\mathbf{X}(i, j)$ would represent the $j$th feature of object $i$. More generally, entry $(i, j) \in [m] \otimes [n]$ of $\mathbf{X}$ is denoted as $\mathbf{X}(i, j)$ or compactly as $x_{ij}$. A matrix is a tensor of order 2.

**Definition 1.2** (Higher Order) A $d$-way tensor is called **higher order** if $d \geq 3$.

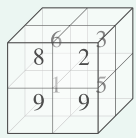
If we have a three-dimensional array of numbers, then we have a **higher-order tensor**. Tensors of order 3 or greater are denoted throughout by uppercase bold Euler roman letters: $\mathcal{X}$. Figure 1.2 shows a vector, a matrix, and an order-3 tensor. If $\mathcal{X}$ is a real-valued tensor of size $m \times n \times p$, then we write $\mathcal{X} \in \mathbb{R}^{m \times n \times p}$. For instance, given a set of $m$ objects, each of which has $n$ features, measured under $p$ different scenarios, the tensor entry $\mathcal{X}(i, j, k)$ would represent the $j$th feature of object $i$ measured in scenario $k$. More generally, entry $(i, j, k) \in [m] \otimes [n] \otimes [p]$ of $\mathcal{X}$ is denoted as $\mathcal{X}(i, j, k)$ or compactly as $x_{ijk}$. We refer to each dimension as a **mode**. We say that mode 1 is of size $m$, mode 2 of size $n$, and mode 3 of size $p$. If all modes have the same size, we call the tensor **cubical**.

A **tensor** is a $d$-way array. We refer to $d$ as the **order** of the tensor and the different ways as **modes**. We say a tensor is **higher-order** if $d \geq 3$.

**Table 1.1** Notation for scalars, vectors, matrices, and higher-order tensors

| Description | Size | Order | Notation | Entry |
|---|---|---|---|---|
| Scalar | 1 | 0 | $x$ | $x$ |
| Vector | $n$ | 1 | $\mathbf{x}$ | $\mathbf{x}(i)$ or $x_i$ |
| Matrix | $m \times n$ | 2 | $\mathbf{X}$ | $\mathbf{X}(i, j)$ or $x_{ij}$ |
| 3-way tensor | $m \times n \times p$ | 3 | $\mathcal{X}$ | $\mathcal{X}(i, j, k)$ or $x_{ijk}$ |
| 4-way tensor | $n_1 \times n_2 \times n_3 \times n_4$ | 4 | $\mathcal{X}$ | $\mathcal{X}(i_1, i_2, i_3, i_4)$ or $x_{i_1 i_2 i_3 i_4}$ |
| $d$-way tensor | $n_1 \times n_2 \times \cdots \times n_d$ | $d$ | $\mathcal{X}$ | $\mathcal{X}(i_1, i_2, \ldots, i_d)$ or $x_{i_1 i_2 \cdots i_d}$ |

**Example 1.1** (Tensor Entries) As an example, consider the $2 \times 2 \times 2$ tensor $\mathcal{X}$, such that

$\mathcal{X} =$  .



It has eight entries:

$$x_{111} = 8, \quad x_{211} = 9, \quad x_{121} = 2, \quad x_{221} = 9,$$
$$x_{112} = 6, \quad x_{212} = 1, \quad x_{122} = 3, \quad x_{222} = 5.$$

**Exercise 1.1** How many entries are in a tensor of size $100 \times 80 \times 60$?

Tensors can go beyond third order. If we have a fourth-order tensor, we begin to run out of letters. So, for a fourth-order tensor, we would likely resort to *subscripts* on the sizes and indices. If $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times n_4}$, then $\mathcal{X}$ is a fourth-order tensor. This is difficult to visualize, but we can think of it as an array of third-order tensors or a matrix of matrices. Its entries are indexed as $(i_1, i_2, i_3, i_4)$ or $x_{i_1 i_2 i_3 i_4}$. For a $d$-way tensor $\mathcal{X}$, its size can be specified as $n_1 \times n_2 \times \cdots \times n_d$, and its entries would be indexed by $d$-tuples of the form $(i_1, i_2, \ldots, i_d) \in [n_1] \otimes [n_2] \otimes \cdots \otimes [n_d]$. In this case, mode 1 is size $n_1$, mode 2 is size $n_2$, and so on. More generally, the size of mode $k \in [d]$ is $n_k$.

**Exercise 1.2** (a) Consider a 3-way tensor of size $512 \times 512 \times 512$. If each entry is a double precision value that requires 8 bytes of memory, how many gigabytes of memory are need for a tensor (note that a gigabyte is $2^{30}$ bytes). (b) What about a 4-way tensor of size $512 \times 512 \times 512 \times 512$?

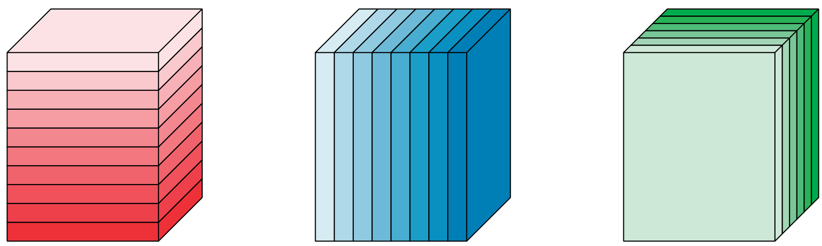
We summarize the notation for tensors in Table 1.1. Because of the awkwardness of tensor notation using many levels of subscripts, this book will generally describe things first in terms of 3-way tensors of size $m \times n \times p$ to establish the concepts, and then generalize to $d$-way tensors of size $n_1 \times n_2 \times \cdots \times n_d$.

## 1.2 Slices and Hyperslices

A slice of a tensor is a 2-way subtensor, which is a matrix. For a third-order tensor, we can give names to all the different 2-way slices.

**Definition 1.3** (Slices of 3-way Tensor) Let $\mathcal{X}$ be a 3-way tensor of size $m \times n \times p$. The $i$th **horizontal slice** is a matrix of size $n \times p$ given by $\mathcal{X}(i, :, :)$. The $j$th **lateral slice** is a matrix of size $m \times p$ given by $\mathcal{X}(:, j, :)$. The $k$th **frontal slice** is a matrix of size $m \times n$ given by $\mathcal{X}(:, :, k)$.

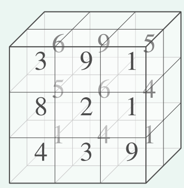
The three types of slices for 3-way tensors are shown in Fig. 1.3. For a tensor of size $m \times n \times p$, the horizontal slices are $\mathcal{X}(i, :, :)$ for all $i \in [m]$ and of size $n \times p$. Likewise, the lateral slices are $\mathcal{X}(:, j, :)$ for all $j \in [n]$ and of size $m \times p$. Finally, the frontal slices are $\mathcal{X}(:, :, k)$ for $k \in [p]$ and of size $m \times n$. The frontal slices can be denoted as $\mathbf{X}_k$ if there is no ambiguity. Tensors are often displayed in terms of their frontal slices.



(a) Horizontal slices, $\mathbf{X}(i, :, :)$. (b) Lateral slices, $\mathbf{X}(:, j, :)$. (c) Frontal slices, $\mathbf{X}(:, :, k)$.

**Figure 1.3** Two-way slices of $10 \times 8 \times 6$ tensor. Dark colors correspond to *higher* indices.

**Example 1.2** (Three-way Tensor Slices) Consider the tensor $\mathcal{X}$ of size $3 \times 3 \times 2$ given by

$\mathcal{X} =$ 


Since its third mode is size 2, it has two frontal slices, each of size $3 \times 3$, i.e., the size of the first two dimensions. So, we can specify $\mathcal{X}$ by listing its frontal slices:

$$\mathcal{X}(:, :, 1) = \begin{bmatrix} 3 & 9 & 1 \\ 8 & 2 & 1 \\ 4 & 3 & 9 \end{bmatrix} \quad \text{and} \quad \mathcal{X}(:, :, 2) = \begin{bmatrix} 6 & 9 & 5 \\ 5 & 6 & 4 \\ 1 & 4 & 1 \end{bmatrix}.$$

The middle horizontal and last lateral slices are

$$\mathcal{X}(2, :, :) = \begin{bmatrix} 8 & 5 \\ 2 & 6 \\ 1 & 4 \end{bmatrix} \quad \text{and} \quad \mathcal{X}(:, 3, :) = \begin{bmatrix} 1 & 5 \\ 1 & 4 \\ 9 & 1 \end{bmatrix}.$$

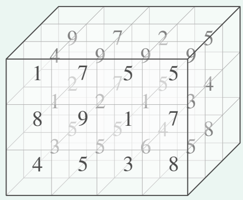
**Exercise 1.3** For the tensor in Example 1.2: (a) What is $\mathcal{X}(1, :, :)$? (b) What is $\mathcal{X}(3, :, :)$? (c) What is $\mathcal{X}(:, 1, :)$? (d) What is $\mathcal{X}(:, 2, :)$?

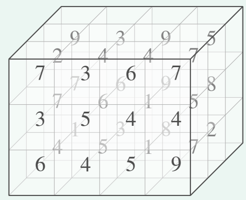
**Exercise 1.4** For the tensor in Example 1.1, list all the (a) horizontal, (b) lateral, and (c) frontal slices.

We can generalize the concept of frontal slices to a tensor of order $d$ for $d > 3$ as follows: A **frontal slice** of a tensor holds every index fixed except the first two. This is convenient for display because the frontal slices are matrices.

**Definition 1.4** (Frontal Slices of $d$-way Tensor) The frontal slices of a $d$-way tensor $\mathcal{X}$ of size $n_1 \times n_2 \times \cdots \times n_d$ are given by $\mathcal{X}(:, :, i_3, i_4, \ldots, i_d)$ for all $(i_3, i_4, \ldots, i_d)$ in $[n_3] \times [n_4] \times \cdots \times [n_d]$.

**Example 1.3** (Frontal Slices) Consider the 4-way $3 \times 4 \times 3 \times 2$ tensor $\mathcal{Y}$ given by

$\mathcal{Y}(:, :, :, 1) =$ 

$, \mathcal{Y}(:, :, :, 2) =$ 

.

The tensor $\mathcal{Y}$ has six frontal slices as follows:

$$\mathcal{Y}(:, :, 1, 1) = \begin{bmatrix} 1 & 7 & 5 & 5 \\ 8 & 9 & 1 & 7 \\ 4 & 5 & 3 & 8 \end{bmatrix}, \quad \mathcal{Y}(:, :, 1, 2) = \begin{bmatrix} 7 & 3 & 6 & 7 \\ 3 & 5 & 4 & 4 \\ 6 & 4 & 5 & 9 \end{bmatrix},$$

$$\mathcal{Y}(:, :, 2, 1) = \begin{bmatrix} 4 & 9 & 9 & 9 \\ 1 & 2 & 1 & 3 \\ 3 & 5 & 6 & 5 \end{bmatrix}, \quad \mathcal{Y}(:, :, 2, 2) = \begin{bmatrix} 2 & 4 & 4 & 7 \\ 7 & 6 & 1 & 5 \\ 4 & 5 & 1 & 7 \end{bmatrix},$$

$$\mathcal{Y}(:, :, 3, 1) = \begin{bmatrix} 9 & 7 & 2 & 5 \\ 2 & 7 & 5 & 4 \\ 5 & 5 & 4 & 8 \end{bmatrix}, \quad \mathcal{Y}(:, :, 3, 2) = \begin{bmatrix} 9 & 3 & 9 & 5 \\ 7 & 6 & 9 & 8 \\ 1 & 3 & 8 & 2 \end{bmatrix}.$$

**Exercise 1.5** Only frontal slices are defined for any order. (a) How many frontal slices does a tensor of size $m \times n \times p \times q$ have? (b) How about a tensor of size $n_1 \times n_2 \times \cdots \times n_d$?

More generally, fixing a single index in an arbitrary-order tensor yields a hyperslice. In other words, we define a **hyperslice** to be the subtensor defined by fixing a single index, and we call this a **mode-$k$ hyperslice**. For example, if $\mathcal{X}$ is a 4-way tensor of size $m \times n \times p \times q$, then the mode-2 hyperslices are $\mathcal{X}(:, j, :, :)$ for all $j \in [n]$. For third-order tensors, mode-1 hyperslices are called horizontal, mode-2 hyperslices are called lateral, and mode-3 hyperslices are called frontal. However, we name the mode-$k$ hyperslices only in the 3-way case.

**Definition 1.5** (Mode-$k$ Hyperslice) The mode-$k$ hyperslice of a $d$-way tensor $\mathcal{X}$ of size $n_1 \times n_2 \times \cdots \times n_d$ is a $(d-1)$-way tensor of size $n_1 \times \cdots \times n_{k-1} \times n_{k+1} \times \cdots \times n_d$. The $j$th mode-$k$ hyperslice is given by $\mathcal{X}(:, \ldots, :, j, :, \ldots, :)$.

**Exercise 1.6** Let $\mathcal{Y}$ be the 4-way $3 \times 4 \times 3 \times 2$ tensor in Example 1.3. (a) What is the size of $\mathcal{Y}(1, :, :, :)$? (b) Write out $\mathcal{Y}(1, :, :, \ell)$ for each $\ell \in \{1, 2\}$. (c) What is the size of $\mathcal{Y}(:, 4, :, :)$? (d) Write out $\mathcal{Y}(:, 4, :, \ell)$ for each $\ell \in \{1, 2\}$. (e) What is the size of $\mathcal{Y}(:, :, 2, :)$? (f) Write out $\mathcal{Y}(:, :, 2, \ell)$ for each $\ell \in \{1, 2\}$.

## 1.3 Tensor Fibers

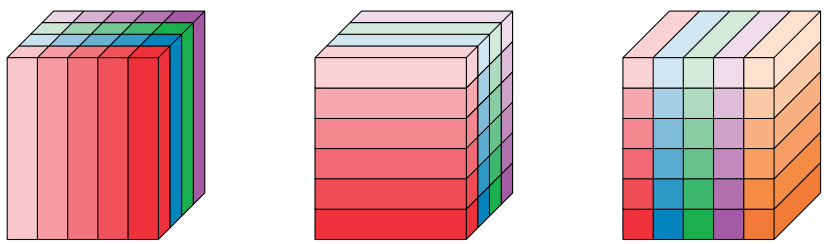
**Tensor fibers** are the generalization of matrix rows and columns.
Tensor fibers are always oriented to be column vectors.

Tensor fibers are the analogs of matrix rows and columns. The main difference between matrix rows and columns and tensor fibers is that tensor fibers are always oriented as column vectors when used in calculations. For a 3-way tensor of size $m \times n \times p$, we have the following:

1. The **mode-1 fibers** of length $m$, also known as **column fibers**, range over all indices in the first mode, holding the second and third indices fixed. In other words, there are $np$ column fibers of the form $\mathbf{x}_{:jk} \in \mathbb{R}^m$.

2. The **mode-2 fibers** of length $n$, also known as **row fibers**, range over all values in the second mode, holding the first and third indices fixed. In other words, there are $mp$ row fibers of the form $\mathbf{x}_{i:k} \in \mathbb{R}^n$.

3. The **mode-3 fibers** of length $p$, also known as **tube fibers**, range over all values in the third mode, holding the first and second indices fixed. In other words, there are $mn$ tube fibers of the form $\mathbf{x}_{ij:} \in \mathbb{R}^p$.



(a) Column fibers, $\mathbf{x}_{:jk}$. (b) Row fibers, $\mathbf{x}_{i:k}$. (c) Tube fibers, $\mathbf{x}_{ij:}$.

**Figure 1.4** Fibers of a third-order tensor of size $6 \times 5 \times 4$.

**Definition 1.6** (Fibers of a 3-way Tensor) Let $\mathcal{X}$ be a 3-way tensor of size $m \times n \times p$. The **column fibers** are vectors of length $m$ given by $\mathcal{X}(:, j, k)$. The **row fibers** are vectors of length $n$ given by $\mathcal{X}(i, :, k)$. The **tube fibers** are vectors of length $p$ given by $\mathcal{X}(i, j, :)$.

The fibers for a third-order tensor are illustrated in Fig. 1.4. More generally, the mode-1 fibers of a third-order tensor $\mathcal{X}$ of size $m \times n \times p$ are given by

$$\mathbf{x}_{:jk} = \begin{bmatrix} x_{1jk} \\ x_{2jk} \\ \vdots \\ x_{mjk} \end{bmatrix} \in \mathbb{R}^m, \qquad \mathbf{x}_{i:k} = \begin{bmatrix} x_{i1k} \\ x_{i2k} \\ \vdots \\ x_{ink} \end{bmatrix} \in \mathbb{R}^n, \qquad \mathbf{x}_{ij:} = \begin{bmatrix} x_{ij1} \\ x_{ij2} \\ \vdots \\ x_{ijp} \end{bmatrix} \in \mathbb{R}^p.$$

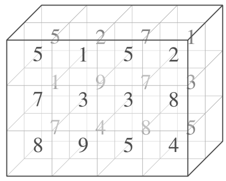
**Exercise 1.7** For an $m \times n \times p$ tensor: (a) How many column fibers are there? (b) How many row fibers? (c) How many tube fibers?

**Example 1.4** (Three-way Tensor Fibers) Consider the 3-way tensor $\mathcal{X}$ defined in Example 1.2. Example mode-1, mode-2, and mode-3 fibers are, respectively,

$$\mathcal{X}(:,2,2) = \mathbf{x}_{:22} = \begin{bmatrix} 9 \\ 6 \\ 4 \end{bmatrix}, \quad \mathcal{X}(1,:,1) = \mathbf{x}_{1:1} = \begin{bmatrix} 3 \\ 9 \\ 1 \end{bmatrix}, \quad \text{and} \quad \mathcal{X}(3,2,:) = \mathbf{x}_{32:} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}.$$

**Exercise 1.8** For the $3 \times 4 \times 2$ tensor $\mathcal{X}$ given below, specify the following fibers: (a) $\mathcal{X}(2,:,2)$, (b) $\mathcal{X}(1,4,:)$, (c) $\mathcal{X}(2,3,:)$, (d) $\mathcal{X}(3,:,1)$, and (e) $\mathcal{X}(:,2,1)$.

$\mathcal{X} =$




.

For a $d$-way tensor, a tensor fiber is a vector extracted from a $d$-way tensor by holding $d-1$ indices fixed. This is analogous to matrix rows and columns. Recall that each column in a matrix ranges over all values in the first dimension, holding the second dimension fixed, while each row in a matrix ranges over all values in the second dimension, holding the first dimension fixed. In general, we say a fiber is a mode-$k$ fiber if all indices are fixed except the $k$th. For a general $d$-way tensor $\mathcal{X}$ of size $n_1 \times n_2 \times \cdots \times n_d$, its **mode-$k$ fibers** are vectors of length $n_k$.

**Definition 1.7** (Mode-$k$ Fiber) A **mode-$k$ fiber** of a tensor is a vector produced by holding all indices but the $k$th fixed.

The concept is straightforward even though the notation is intricate:

$$\mathcal{X}(i_1,\ldots,i_{k-1},:,i_{k+1},\ldots,i_d) = \begin{bmatrix} \mathcal{X}(i_1,\ldots,i_{k-1},\ 1\ ,i_{k+1},\ldots,i_d) \\ \mathcal{X}(i_1,\ldots,i_{k-1},\ 2\ ,i_{k+1},\ldots,i_d) \\ \vdots \\ \mathcal{X}(i_1,\ldots,i_{k-1},n_k,i_{k+1},\ldots,i_d) \end{bmatrix} \in \mathbb{R}^{n_k}.$$

**Example 1.5** (Four-way Tensor Fibers) Some example fibers from the 4-way tensor in Example 1.3 are as follows:

$$\text{Mode 1: } \mathcal{Y}(:,2,1,1) = \begin{bmatrix} 7 \\ 9 \\ 5 \end{bmatrix}, \qquad \text{Mode 2: } \mathcal{Y}(3,:,3,1) = \begin{bmatrix} 5 \\ 5 \\ 4 \\ 8 \end{bmatrix},$$

$$\text{Mode 3: } \mathcal{Y}(2,3,:,1) = \begin{bmatrix} 1 \\ 1 \\ 5 \end{bmatrix}, \qquad \text{Mode 4: } \mathcal{Y}(1,2,3,:) = \begin{bmatrix} 7 \\ 3 \end{bmatrix}.$$

**Exercise 1.9** Let $\mathcal{Y}$ be the 4-way tensor in Example 1.3. Specify the following fibers: (a) $\mathcal{Y}(:,1,3,2)$, (b) $\mathcal{Y}(1,:,2,1)$, (c) $\mathcal{Y}(3,3,:,1)$, and (d) $\mathcal{Y}(1,2,2,:)$.

**Exercise 1.10** For an $m \times n \times p \times q$ tensor: (a) How many mode-1 fibers are there? (b) How many mode-2 fibers? (c) How many mode-3 fibers? (d) How many mode-4 fibers?

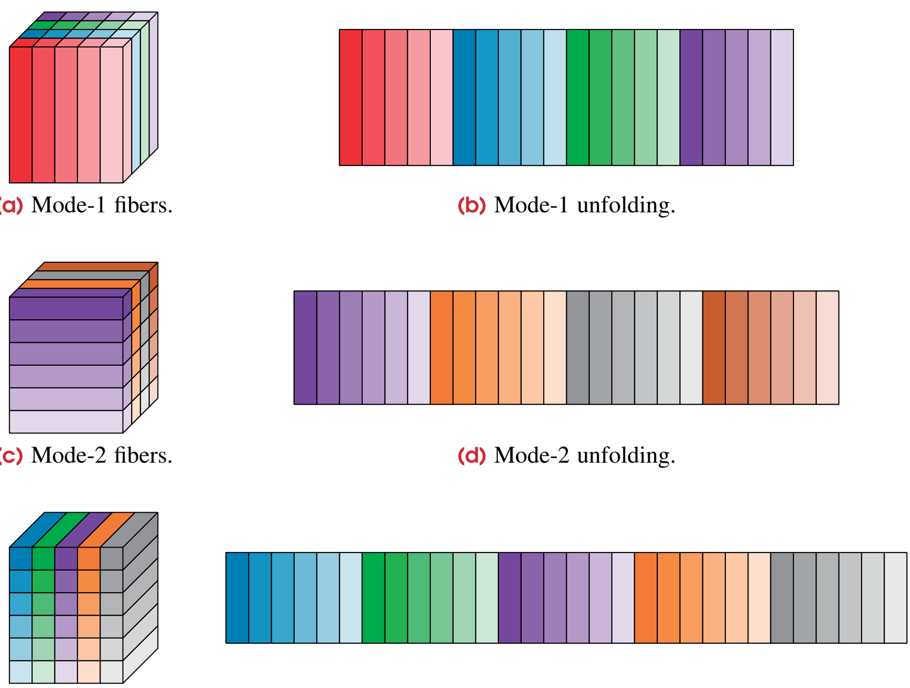
**Exercise 1.11** For a tensor of size $n_1 \times n_2 \times \cdots \times n_d$, how many mode-$k$ fibers are there?

## 1.4 Tensor Mode-$k$ Unfolding

The elements of a tensor can be rearranged to form various matrices in a procedure referred to as **unfolding**, also known as **matricization**. A particular unfolding of interest is the mode-$k$ unfolding defined as follows.

**Definition 1.8** (Informal Definition of Mode-$k$ Unfolding) The **mode-$k$ unfolding** of a tensor is a matrix whose columns are the mode-$k$ fibers of that tensor, denoted as $\mathbf{X}_{(k)}$.

We defer the precise definitions (which explain how the columns are ordered) until Section 2.3. We illustrate the mode-$k$ unfoldings of a 3-way tensor in Fig. 1.5.



(a) Mode-1 fibers. (b) Mode-1 unfolding.

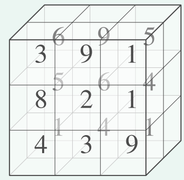
(c) Mode-2 fibers. (d) Mode-2 unfolding.

(e) Mode-3 fibers. (f) Mode-3 unfolding.

**Figure 1.5** Illustration of mode-$k$ unfoldings of a 3-way tensor.

**Example 1.6** (Tensor Unfolding) Consider the tensor $\mathcal{X}$ of size $3 \times 3 \times 2$ from Example 1.2:

$\mathcal{X} =$




.

Its mode-1 and mode-2 unfoldings are

$$\mathbf{X}_{(1)} = \begin{bmatrix} 3 & 9 & 1 & 6 & 9 & 5 \\ 8 & 2 & 1 & 5 & 6 & 4 \\ 4 & 3 & 9 & 1 & 4 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{X}_{(2)} = \begin{bmatrix} 3 & 8 & 4 & 6 & 5 & 1 \\ 9 & 2 & 3 & 9 & 6 & 4 \\ 1 & 1 & 9 & 5 & 4 & 1 \end{bmatrix}.$$

Finally, its mode-3 unfolding is

$$\mathbf{X}_{(3)} = \begin{bmatrix} 3 & 8 & 4 & 9 & 2 & 3 & 1 & 1 & 9 \\ 6 & 5 & 1 & 9 & 6 & 4 & 5 & 4 & 1 \end{bmatrix}.$$

**Exercise 1.12** Let $\mathcal{X}$ be a tensor of size $10 \times 8 \times 6$. What is the size of $\mathbf{X}_{(2)}$?

## 1.5 Example Tensors

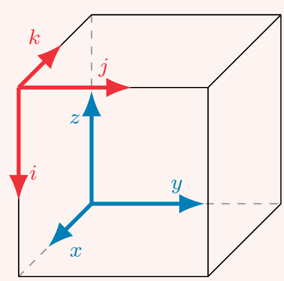
We describe several tensors from real-world datasets to help us understand the prevalence of tensor-formatted data. These examples will be used throughout the book. As much as possible, we visualize the data in tensor format so that we can see the connection between the data and its representation as a tensor.

### 1.5.1 Miranda Scientific Simulation Data

Computational fluid dynamics uses numerical simulations to understand the flow of liquids or gases, with ubiquitous applications ranging from combustion engines to aerodynamics of aircraft wings to weather prediction. Mathematically, the problem can be solved using discretized partial differential equations. Direct numerical simulation is a technique that solves fluid flow problems on a uniform Cartesian grid, stepping through time. The datasets are massive since an $n \times n \times n$ Cartesian grid generates $n^3$ data for each timestep, resulting in terabytes of data from even modest sized simulations.

***Remark* 1.9** (Tensor versus Cartesian indexing)

Tensors are indexed starting in the front upper left corner, with the first index corresponding to the downward vertical direction, the second index corresponding to the horizontal direction, and the third index corresponding to the backward lateral direction. In contrast, Cartesian coordinates start in the back lower left corner, with the first index corresponding to the lateral direction, the second index corresponding to the horizontal direction, and the third index corresponding to the vertical direction.



Our data comes originally from Cabot and Cook (2006) via the Scientific Data Reduction Benchmark (SDRBench) of Zhao et al. (2020). The simulation is a Rayleigh–Taylor instability direct numerical simulation of the mixing of two fluids of different densities. The calculation produces density measurements over time on a 3D uniform Cartesian spatial grid of size $3072 \times 3072 \times 3072$. In single precision, the density measurements from a single timestep requires more than 13 GB of storage.

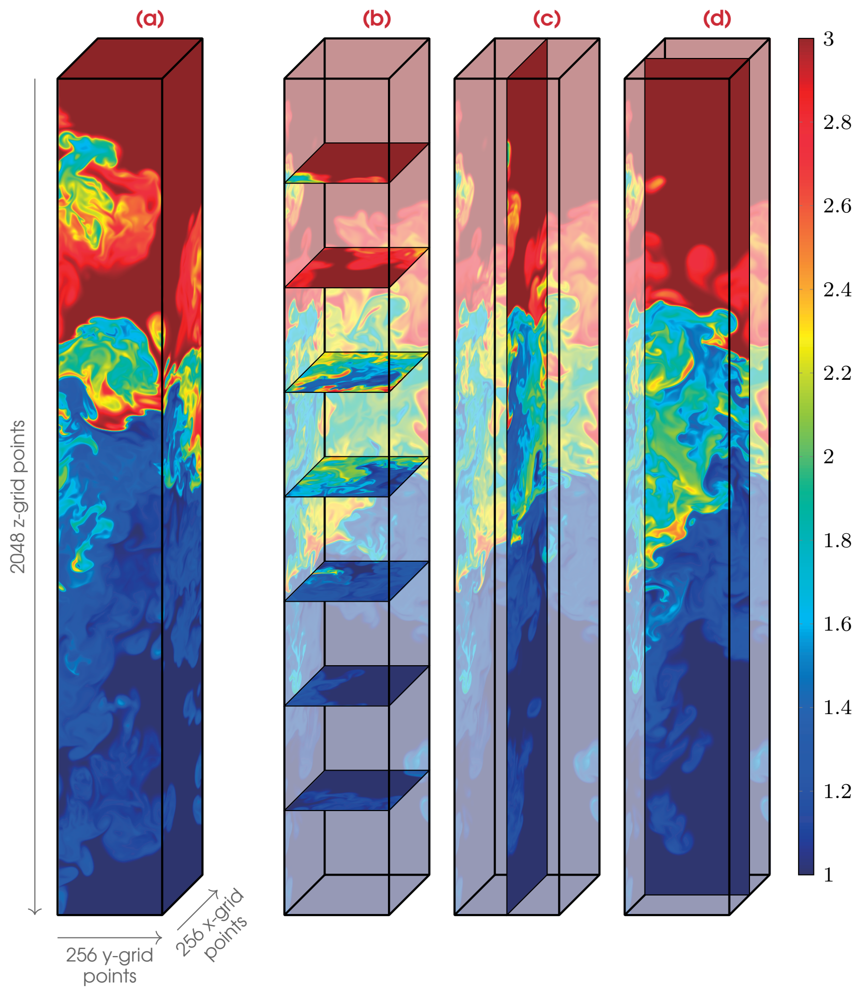



**Figure 1.6** Miranda tensor of size $2048 \times 256 \times 256$, capturing the mixing of two fluids of different densities. Color indicates density. (a) Outermost slices: the horizontal slice at $i = 1$, the lateral slice at $j = 256$, and the frontal slice at $k = 1$. (b) Horizontal slices at $i = \{256, 512, \ldots, 1792\}$. (c) Lateral slice at $j = 128$. (d) Frontal slice at $k = 128$.

Our specific dataset is from a single time point and, in order to keep the memory requirements manageable, uses only a subset of the full spatial grid. Additionally, we remap the Cartesian coordinates to tensor coordinates per Remark 1.9. The resulting tensor is of size

$$2048 \text{ z-grid points} \quad \times \quad 256 \text{ y-grid points} \quad \times \quad 256 \text{ x-grid points}.$$

This tensor requires 1 GB of storage in double precision. The data is available for download (approximately 300 MB of lossless compressed storage) at https://gitlab.com/tensors/tensor_data_miranda_sim (Ballard et al., 2022).

Simulation data on a regular grid can be represented as a tensor.

We visualize the Miranda tensor in Fig. 1.6. The top horizontal slice is purely the high-density fluid (density $= 3$) and the bottom horizontal slice is purely the low-density fluid (density $= 1$), and the mixing happens in between. We show middle slices in each mode.

**Exercise 1.13** Using slice notation, what are the 2D matrices being visualized in Fig. 1.6?

### 1.5.2 EEM Fluorescence Spectroscopy Data

In fluorescence spectroscopy, a chemical sample is excited, and the light that is emitted is measured at several different wavelengths, resulting in an excitation–emission matrix (EEM) of fluorescence intensities. When EEM data is gathered for a number of samples, we obtain a 3-way tensor. This data can be used in analytical chemistry for estimating chemical compound concentrations and spectra from multiple mixtures. It has applications, for example, in environmental modeling. See Smilde et al. (2004, chapter 10.2) for further details.

Our specific EEM example data has been curated from a series of fluorescence spectroscopy experiments as reported by Acar et al. (2014). The data has been preprocessed to fill in missing data and replace negative entries as explained in the `README` file of the data repository. (We revisit the raw data in our discussion of handling missing data in Chapter 14.) All the entries are nonnegative. The data is available for download at https://gitlab.com/tensors/tensor_data_eem (Kolda, 2021a).

The data comprises EEM measurements on 18 samples, each of which is a mixture of three chemical compounds:

- valine–tyrosine–valine (Val-Tyr-Val), a peptide,
- tryptophan–glycine (Trp-Gly), a peptide,
- phenylalanine (Phe), an amino acid.

The intensities are measured at 251 emission wavelengths $(250, 251, \ldots, 500$ nm) and 21 excitation wavelengths $(210, 215, \ldots, 310$ nm). The 18 EEM profiles are shown in Fig. 1.8 as surface plots. The first three samples contain only a single compound, and each compound creates a peak (a bright spot) centered at a different point. (These samples can be removed to make the analysis more interesting.) Samples 4–18 are mixtures of the compounds, so their profiles are, in a sense, weighted combinations of the first three. For instance, the last mixture is a mixture of 3.75 parts Val-Tyr-Val and 5.00 parts Phe, so it can be viewed as a weighted combination of the first and third profiles.

Stacking the emission–excitation intensity profile matrices yields the 3-way EEM tensor of size

$$18 \text{ samples} \quad \times \quad 251 \text{ emissions} \quad \times \quad 21 \text{ excitations}.$$

It is illustrated in Fig. 1.7.

Each sample produces an emission–excitation matrix, and the EEM data from multiple samples is combined to form the EEM tensor.

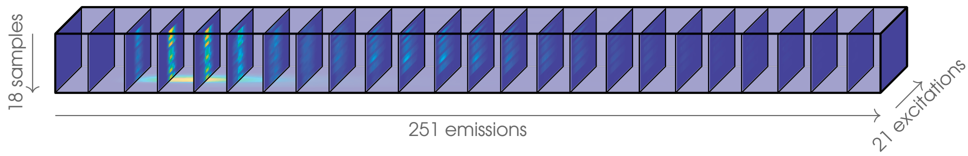



**Figure 1.7** EEM tensor of size $18 \times 251 \times 21$, highlighting a selection of lateral slices.

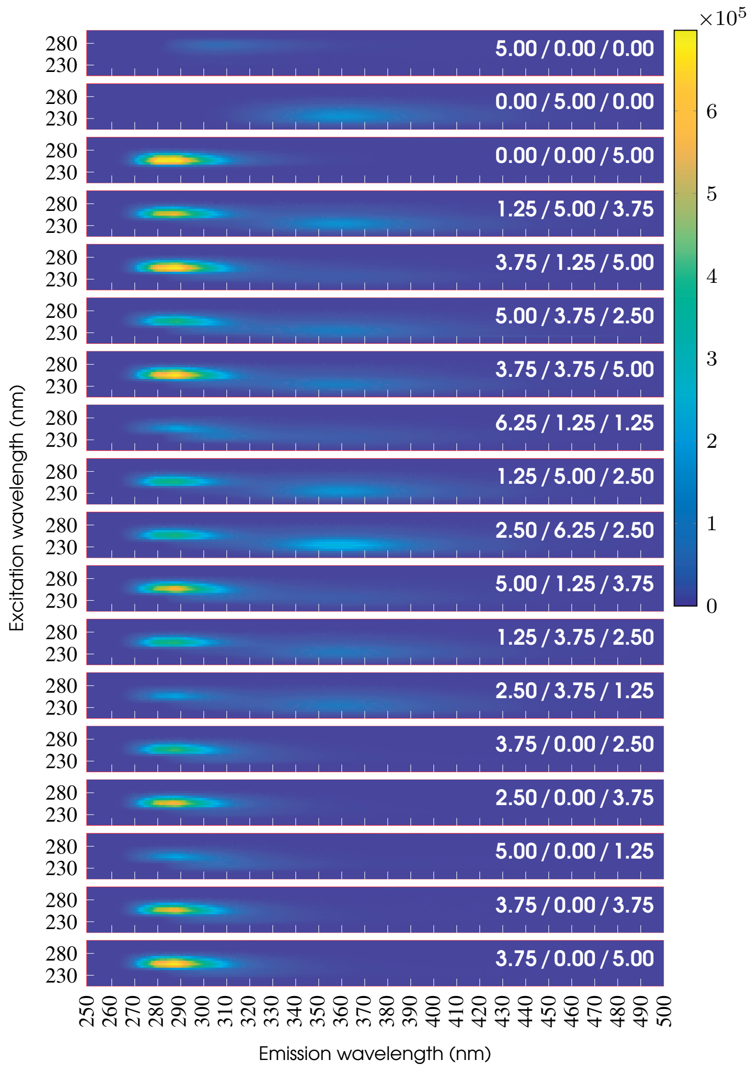



**Figure 1.8** Emission–excitation intensity profiles of EEM tensor. The profiles correspond to horizontal slices of the EEM tensor, ordered from top ($\mathcal{X}(1, :, :)$) to bottom ($\mathcal{X}(18, :, :)$). Each profile is labeled at right with concentrations of three chemical compounds (Val-Tyr-Val / Trp-Gly / Phe). Each profile covers 21 excitation wavelengths (210, 215, …, 310 nm) by 251 emission wavelengths (250, 251, …, 500 nm).

### 1.5.3 Monkey BMI Neuronal Spike Data

We consider a dataset of monkey (Rhesus macaque) behavior using a brain–machine interface (BMI) in a series of trials. The monkey BMI tensor data has been curated from a series of experiments as reported in Vyas et al. (2018, 2020) and Williams et al. (2018) and is available for download at https://gitlab.com/tensors/tensor_data_monkey_bmi (Kolda, 2022a). In each of 88 experiments, a monkey moves a cursor to one of four targets (at 0, 90, 180, and 270 degrees) and holds it there for 500 ms using a BMI (Fig. 1.9).

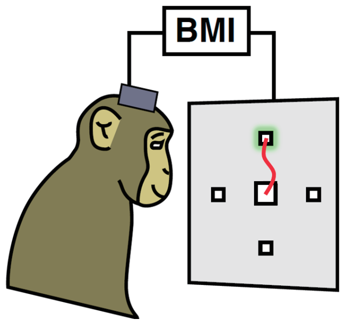



**Figure 1.9** The BMI task is to move the cursor from the center to one of four targets.

During this task, neuron spike data is collected. The time per trial varies, but we have standardized every trial to 200 timesteps. Specifically, the data has been time-aligned so that $t = 0$ is the start, $t = 100$ is time of target acquisition, and $t = 200$ is the end after 500 ms of holding the cursor at the target. The data has been additionally preprocessed to smooth the spikes, remove trials for which target acquisition took more than 600 ms, and remove neurons with little to no activity. The neurons are sorted by level of activity, from greatest to least. The resulting tensor is

$$43 \text{ neurons} \quad \times \quad 200 \text{ timesteps} \quad \times \quad 88 \text{ trials}.$$

The tensor is shown in Fig. 1.10.

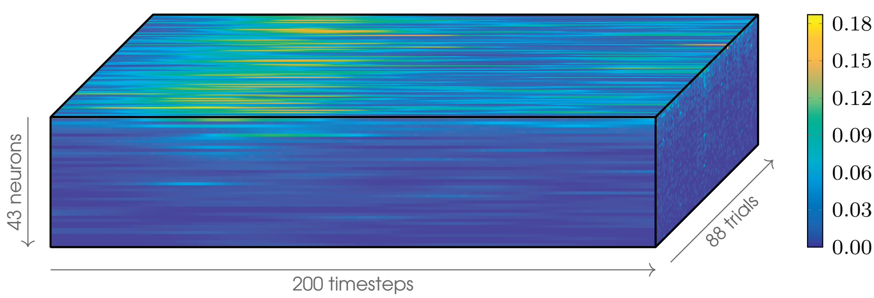



**Figure 1.10** Monkey BMI tensor of size $43 \times 200 \times 88$.

The 88 trials are split among the four targets as shown in Table 1.2.

The activities of several neurons across the trials are shown in Fig. 1.11. Each subimage corresponds to a horizontal slice of the tensor, i.e., $\mathcal{X}(i,:,:)$ shows the activities of neuron $i$. Within each figure, the individual lines correspond to tensor row fibers, i.e., $\mathcal{X}(i,:,k)$ is

**Table 1.2** Number of trials for each angle in the monkey BMI tensor

| **Angle** | 0 | 90 | 180 | 270 |
|---|---|---|---|---|
| **Count** | 20 | 28 | 21 | 19 |

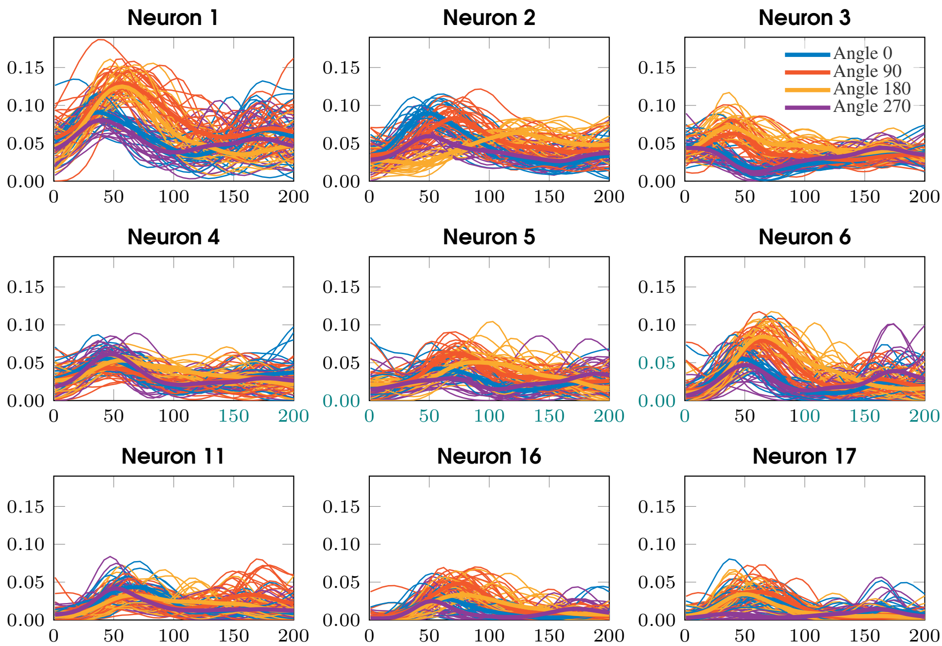



**Figure 1.11** Activities of example neurons. For each neuron, thin lines correspond to activity in each of 88 trials, color-coded by the target angle. Thick lines are averages. Times 1–100 are target acquisition, and times 101–200 are holding the cursor at the target.

the activity of neuron $i$ in trial $k$. The lines are color-coded according to the target. For each target, the average for all trials is shown as a thick line.

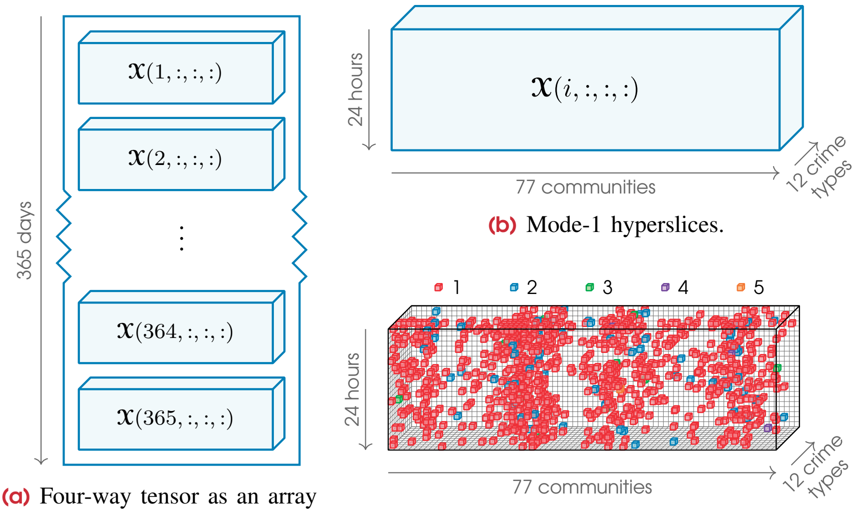
The recording from a single neuron in a single trial is a vector of observations over time; the recordings of all neurons from a single trial forms a matrix; and the collection of all (time-normalized) trials forms the monkey BMI tensor.

**Exercise 1.14** Which type of fiber (row, column, or tube) corresponds to the reading of an individual neuron from a single trial?

## 1.5.4 Chicago Crime Count Data

The Chicago crime data is statistics from public safety criminal activity reports in the city of Chicago. The data is available at www.cityofchicago.org, and we are using a 4-way tensor version corresponding to a single year of data, available at `https://gitlab.com/tensors/tensor_data_chicago_crime` in file `chicago_crime_2019.mat` (Kolda, 2022b).

The tensor modes correspond to 365 days (January 1 through December 31, 2019), 24




**(a)** Four-way tensor as an array of 3-way tensors.

**(b)** Mode-1 hyperslices.

**(c)** Sparsity pattern of $\mathcal{X}(1,:,:,:)$.

**Figure 1.12** Chicago crime tensor of size $365 \times 24 \times 77 \times 12$.

hours, 77 communities, and 11 crime types. Entry $\mathcal{X}(i, j, k, \ell)$ is the number of times that crime $\ell$ happened in neighborhood $k$ during hour $j$ on day $i$. Hence, the tensor is formatted as

$$365 \text{ days} \quad \times \quad 24 \text{ hours} \quad \times \quad 77 \text{ communities} \quad \times \quad 12 \text{ crime-types}.$$

We have treated time as two-dimensional, splitting hours and days into two modes in order to expose daily patterns in addition to longer-term trends. We can visualize the 4-way tensor as an array of 3-way tensors as in Fig. 1.12a, and each 3-way subtensor is formatted as in Fig. 1.12b.

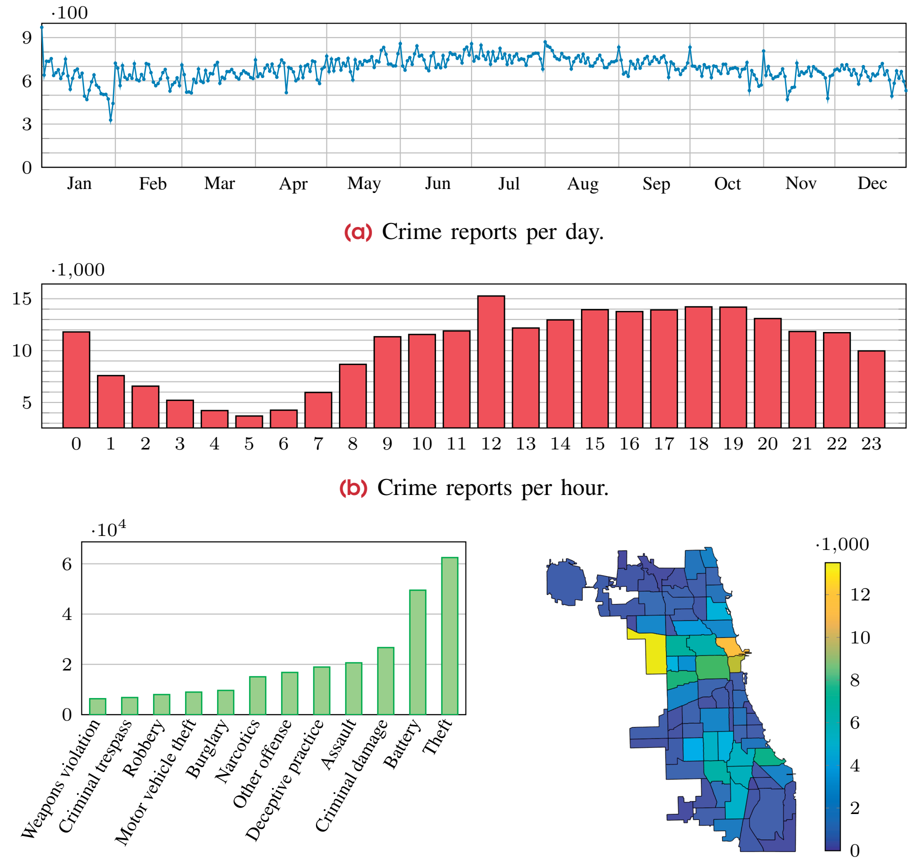
> Time can be multidimensional. For example, hourly data can be divided further into days, weeks, etc.

The tensor is **sparse** because it has only 230,591 nonzeros out of 8,094,240 entries; that is, only 2.85% of its entries are nonzero. Storing $\mathcal{X}$ as a sparse tensor (i.e., storing each nonzero and 4-tuple index) requires less than 15% of the storage of the dense tensor. To visualize the sparsity, consider the first mode-1 hyperslice, $\mathcal{X}(1,:,:,:)$, pictured in Fig. 1.12c. It has only 861 nonzeros out of 22,176 entries.

We compute some statistics on the Chicago crime tensor. The number of crime reports per day are shown in Fig. 1.13a. Crime reports are highest overall in the summer months, with a peak during August 1–4, which so happens to correspond to the Lollapalooza 2019 festival in Grant Park. The day with the most reports overall is January 1, 2019, keeping in mind that various factors affect the date of a crime report and the first of the year is presumably a popular day to choose when the exact date is uncertain.

Figure 1.13b shows the cumulative crimes per hour, with hour 0 corresponding to midnight to 12:59 a.m., and hour 23 corresponding to 11:00–11:59 p.m.. Crime reports are lowest in the hours 1:00–7:00 a.m. and peak at noon.

Totals crime reports per type are listed in Fig. 1.13c. The preprocessing of the data removed




**(a)** Crime reports per day.

**(b)** Crime reports per hour.

**(c)** Crime reports by type.

**(d)** Crime reports by neighborhood.

**Figure 1.13** Chicago crime report counts from January 1, 2019 to December 31, 2019.

any crimes that occurred fewer than 5000 times in the time period of the data. The crimes are in order of overall prevalence, with theft corresponding to index 1, battery to index 2, and so on down to weapons violation corresponding to index 12.

A heatmap of the total crime frequency (over all crime types) per community is shown in Fig. 1.13d. This is not normalized by population. The majority of reports come from the community area known as Austin in the West Side region of Chicago.

**Exercise 1.15** Load the tensor data and recreate Figs. 1.13a and 1.13b.

## 1.6 A First Look at Tensor Decompositions

The goal of this book is to learn how to **decompose** tensors into representations that might be smaller, more expressive, or some combination of these ideals. Like matrix decompositions, we seek a set of matrices/tensors that can be multiplied together appropriately to reconstruct the input. Unlike matrix decompositions, tensor decompositions are rarely exact representations and instead only approximations of the input. Most tensor decompositions can be viewed as generalizations of low-rank matrix approximations to higher-order data. We focus on two types of decompositions, Tucker and CP; we more briefly discuss other decompositions in Chapters 8 and 17.

### 1.6.1 A First Look at Tucker Decomposition

A **Tucker decomposition** *compresses* a tensor by decomposing into a smaller **core tensor** multiplied by a matrix in each mode.

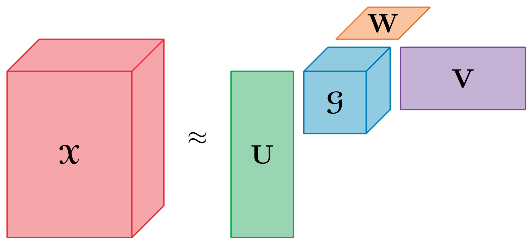



Figure 1.14 Tucker decomposition.

We visualize the 3-way case as in Fig. 1.14. Here $\mathcal{X}$ is the original 3-way tensor, $\mathcal{G}$ is the core 3-way tensor, and the matrices $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$ are the matrices that are multiplied with $\mathcal{G}$ to approximate $\mathcal{X}$. The tensor $\mathcal{G}$ can be interpreted as a compressed version of $\mathcal{X}$, and the matrices $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$ are bases for the subspaces onto which $\mathcal{X}$ is projected for compression.

In Tucker, it is possible to choose the size of the compressed tensor to ensure that the approximation error is below a user-specified error threshold. The challenge in the Tucker decomposition is identifying the optimal subspaces for compression.

### 1.6.2 A First Look at CP Decomposition

A **CP decomposition** expresses a tensor as a sum of vector outer products. The summands are called **components**. The vectors in the components are used for *interpretation*.

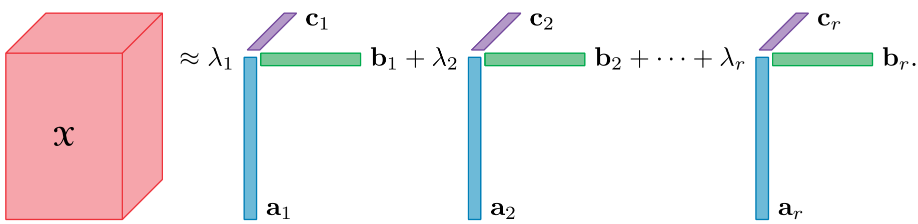



Figure 1.15 CP decomposition.

We visualize the 3-way CP decomposition in Fig. 1.15. Each component is the outer product of three vectors. The vectors constituting the components are usually explanatory as to the nature of the component.

In comparison to Tucker, CP decomposition is often viewed as more useful for interpretation. The challenges in CP decomposition are choosing an appropriate number of components and computing the optimal solution.