Examples of matrices from data science

COMS 3251 Fall 2022 (Daniel Hsu)

1 Topic matrix

Suppose you gather statistics on how often each word in the English language (a vocabulary of $m \approx 5 \times 10^4$ words) is used in various categories of internet "news" articles.

- In "politics", you expect words like "government" and "president" to show up a lot.
- In "science", it is words like "hypothesis" and "experiment" that will be quite frequent.

For each category ("politics", "science", "sports", etc.), create an m-vector whose ith component is the proportion of words in articles from the category (strung together as a single long sequence of words) that are equal to the ith vocabulary word. Suppose the first vocabulary word is "hypothesis"; it accounts for $1.3/10^4$ fraction of words across "science" articles, but only a $2.2/10^6$ fraction of words across "politics" articles. Suppose the second vocabulary word is "freedom"; it accounts for a $3.1/10^4$ fraction of words across "politics" articles, and a $5.8/10^5$ fraction of words across "science" articles. And so on:

$$\mathbf{a}_{\text{politics}} = \begin{bmatrix} 2.2/10^6 \\ 3.1/10^4 \\ \vdots \end{bmatrix}, \quad \mathbf{a}_{\text{science}} = \begin{bmatrix} 1.3/10^4 \\ 5.8/10^5 \\ \vdots \end{bmatrix}, \quad \mathbf{a}_{\text{sports}} = \begin{bmatrix} 4.0/10^6 \\ 4.0/10^6 \\ \vdots \end{bmatrix}, \quad \dots .$$

The matrix A whose columns are these m-vectors for the various categories is a "topic matrix":

$$A = \begin{bmatrix} \uparrow & \uparrow & \uparrow & \\ \mathbf{a}_{\text{politics}} & \mathbf{a}_{\text{science}} & \mathbf{a}_{\text{sports}} & \cdots \\ \downarrow & \downarrow & \downarrow & \end{bmatrix}.$$

Such matrices are frequently used in information retrieval and natural language processing. (There are many other ways to get a topic matrix.)

What are the word frequencies in "political science"? This category is not represented in our topic matrix. Let's assume that "political science" is 60% "politics" and 40% "science". Multiply the topic matrix A by the vector $\mathbf{x} = (0.6, 0.4, 0, \ldots, 0)$:

$$\mathbf{b} = \underbrace{\begin{bmatrix} \uparrow & \uparrow & \uparrow & \\ \mathbf{a}_{\text{politics}} & \mathbf{a}_{\text{science}} & \mathbf{a}_{\text{sports}} & \cdots \\ \downarrow & \downarrow & \downarrow & \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} 0.6 \\ 0.4 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\mathbf{x}} = 0.6\,\mathbf{a}_{\text{politics}} + 0.4\,\mathbf{a}_{\text{science}}.$$

The first two components of $\mathbf{b} = (b_1, \ldots, b_m)$ are

$$\begin{aligned} b_1 &= 0.6 \times {}^{2.2}/_{10^6} + 0.4 \times {}^{1.3}/_{10^4} \approx {}^{5}/_{10^5} \quad \text{(frequency of "hypothesis")}, \\ b_2 &= 0.6 \times {}^{3.1}/_{10^4} + 0.4 \times {}^{5.8}/_{10^5} \approx {}^{2}/_{10^4} \quad \text{(frequency of "freedom")}. \end{aligned}$$

You expect the word "hypothesis" to be about ${}^{5}/_{0.22} \approx 23$ times more prevalent in "political science" than it is in "politics".

2 Downsampling

Suppose an n-vector $\mathbf{x} = (x_1, \ldots, x_n)$ represents a time series, which is a list of time-ordered data points. So x_1 is the data point associated with the first time point, x_2 with the second time point, and so on. The time points, which need to be agreed upon to interpret the time series, are often chosen to be equally spaced apart. For example, $\mathbf{x}$ could represent a 1 second audio signal, and each data point may be the sound pressure (in pascals) sampled every $1/n$ fraction of a second.

Downsampling is a method to reduce the "size" of a time series. For example, we may convert a time series $\mathbf{x}$ for a 1 second audio signal sampled every $1/n$ fraction of a second into a time series $\mathbf{y}$ for the same signal but sampled every $2/n$ fraction of a second. To achieve this, we multiply an

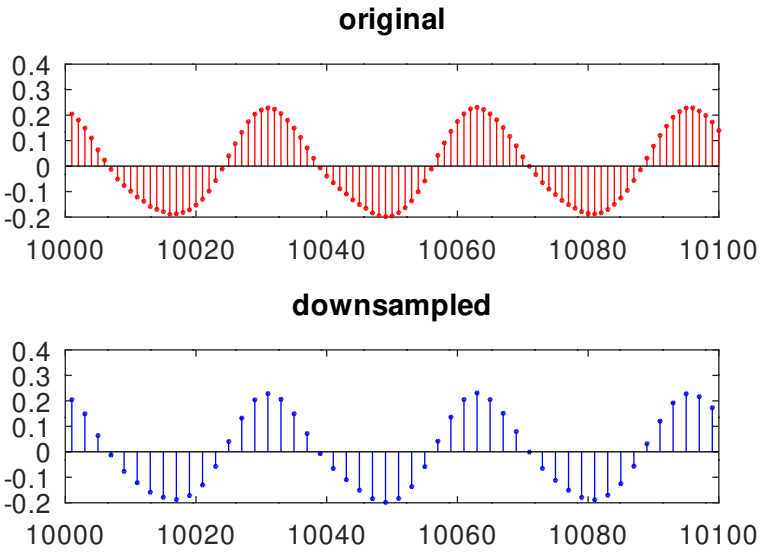

Figure 1: Plot of (a 0.0125 second portion of) an audio recording (top) and its downsampled version (bottom). The original recording was sampled at 8000 hertz, which means each time unit on the horizontal axes represents a 1/8000 fraction of a second.

appropriate downsampling matrix A by $\mathbf{x}$:

$$\mathbf{y} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 1 & 0 & \cdots \\ & & & \vdots & & & \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ \vdots \end{bmatrix}}_{\mathbf{x}} = \begin{bmatrix} x_1 \\ x_3 \\ x_5 \\ \vdots \end{bmatrix}. \tag{1}$$

Some data in $\mathbf{x}$ are now lost after downsampling, but the vector $\mathbf{y}$ has half as many components. See Figure 1 for an example.

Downsampling can also be used for spatial signals, such as a 2D images. A 2D gray-scale image can be represented as a matrix of pixel intensity values:

$$\begin{bmatrix} x_{1,1} & \cdots & x_{1,w} \\ \vdots & \ddots & \vdots \\ x_{h,1} & \cdots & x_{h,w} \end{bmatrix} = \begin{bmatrix} \uparrow & & \uparrow \\ \mathbf{x}_1 & \cdots & \mathbf{x}_w \\ \downarrow & & \downarrow \end{bmatrix},$$

but we can reorganize the matrix into an (hw)-vector $\mathbf{x}$ by stacking the

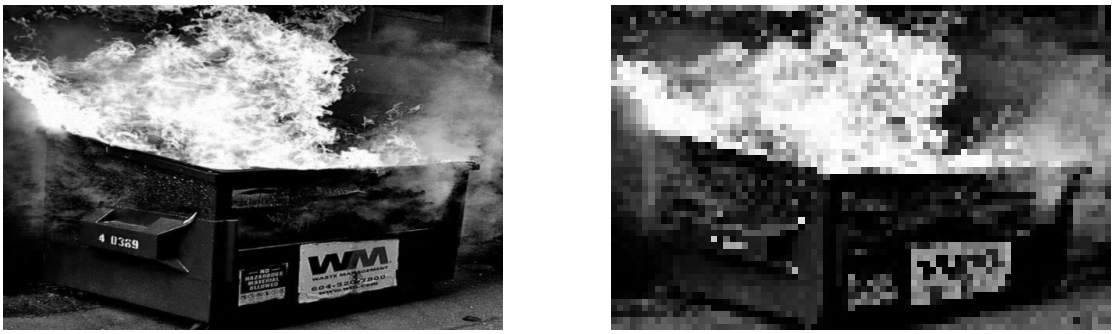

Figure 2: A 416×640 gray-scale image (left) and its 52×80 downsampled version (right). Each pixel of the image on the right is rendered in a way to be as large as an 8×8 block of pixels from the image on the left.

columns:

$$\mathbf{x} \;=\; \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_w \end{bmatrix}.$$

To downsample the $h \times w$ image $\mathbf{x}$ to get an $(h/2) \times (w/2)$ image, we apply the downsampling matrix A from (1) (with $n = h$) to each of $\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_5, \ldots$:

$$\begin{bmatrix} A & O & O & O & O & O & \cdots \\ O & O & A & O & O & O & \cdots \\ O & O & O & O & A & O & \cdots \\ & & & \vdots & & & \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \end{bmatrix} \;=\; \begin{bmatrix} A\mathbf{x}_1 \\ A\mathbf{x}_3 \\ A\mathbf{x}_5 \\ \vdots \end{bmatrix}, \tag{2}$$

where O is the all-zeros matrix. See Figure 2 for an example (where an $h \times w$ image is downsampled to an $(h/8) \times (w/8)$ image).

In an actual application, there is no need to explicitly construct the matrix A from (1) (or the matrix from (2)). One is likely better off implementing the downsampling function directly. But the representation of the downsampling function as matrices is helpful for reasoning about the function. For one, it becomes clear that downsampling is a linear function. So to downsample the average of two images, we can average the downsampled versions of the two images. Another benefit is that the representations in (1) and (2) may help one understand how to modify the function. Suppose you want to reduce an

$h \times w$ image to an $(h/2) \times w$ image. How should you change (2)? What if you want to get an $h \times (w/2)$ image?

3 Moving averages

Some time series have short-term fluctuations that may obscure longer-term trends. To help bring out the longer-term trends, one may try to "smooth out" the short-term fluctuations using moving averages. Given an n-vector $\mathbf{x} = (x_1, \ldots, x_n)$ representing a time series, the 3-period moving average yields a $(n-2)$-vector $\mathbf{y} = (y_1, \ldots, y_{n-2})$ such that

$$y_i \;=\; \frac{x_i + x_{i+1} + x_{i+2}}{3}.$$

This is realized by the matrix-vector multiplication $\mathbf{y} = A\mathbf{x}$ as follows:

$$\mathbf{y} \;=\; \underbrace{\begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 1/3 & 1/3 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1/3 & 1/3 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1/3 & 1/3 & 1/3 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{n-3} \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix}}_{\mathbf{x}} \;=\; \begin{bmatrix} (x_1+x_2+x_3)/3 \\ (x_2+x_3+x_4)/3 \\ \vdots \\ (x_{n-3}+x_{n-2}+x_{n-1})/3 \\ (x_{n-2}+x_{n-1}+x_n)/3 \end{bmatrix}.$$

This naturally generalizes to k-period moving averages for positive integers $k \in \{1, \ldots, n\}$. The larger k is, the smoother the resulting time series is. See Figure 3 for an example.

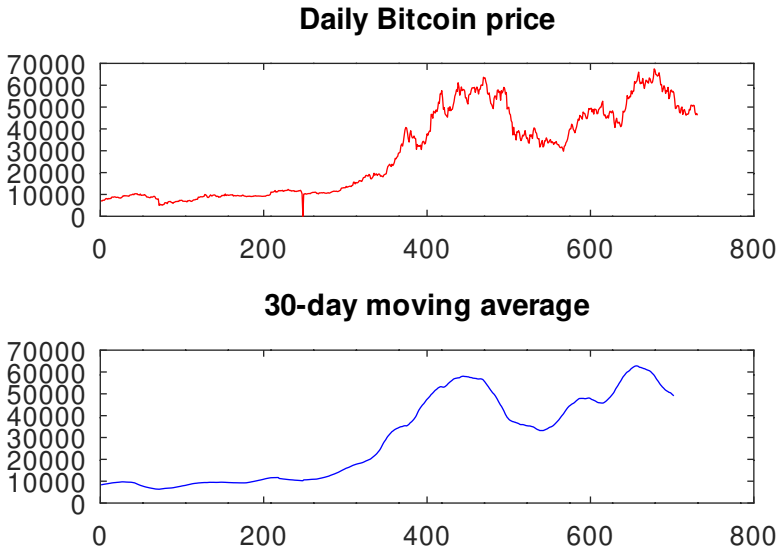

Figure 3: Plot of the daily Bitcoin price (in US dollars) between January 1, 2020 and December 31, 2021 (top) and its 30-day moving average (bottom).