CS 4780/6780: Fundamentals of Data Science **Spring 2019**

Topic 4: Review of Linear Algebra

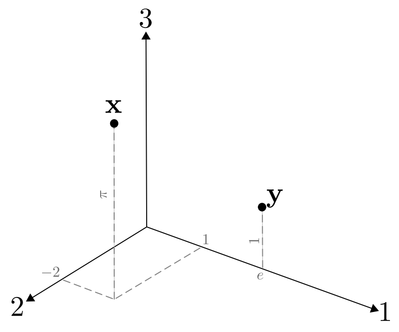
Instructor: Daniel L. Pimentel-Alarcón

4.1 Vectors

In words, a *vector* is simply a point in space.

Example 4.1. Here are two vectors in 3-dimensional space, often denoted as $\mathbb{R}^3$:

$$\mathbf{x} = \begin{bmatrix} 1 \\ -2 \\ \pi \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} e \\ 0 \\ 1 \end{bmatrix}.$$



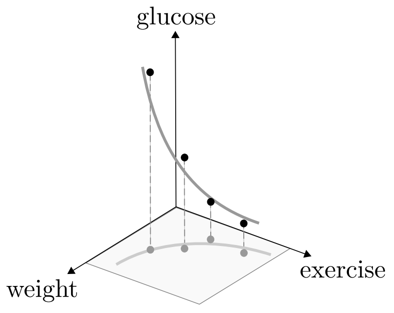
Why do I care about vectors?

If you are wondering this, you are asking yourself the right question! The reason is simple: in data science we have to deal with data, and it is useful to arrange it in vectors.

Example 4.2 (Electronic health records)**.** Hospitals keep health records of their patients, containing information such as weight, amount of exercise they do, and glucose level. The information of the i$^{\text{th}}$ patient can be arranged as a vector

$$\mathbf{x}_{\rm i} = \begin{bmatrix} \text{weight} \\ \text{exercise} \\ \text{glucose} \end{bmatrix} \in \mathbb{R}^3.$$

In this sort of problem we want to identify *causes* for diseases. This can be done by analyzing the patterns in the vectors of different patients. For example, if our data $\mathbf{x}_1, \ldots, \mathbf{x}_N$ looks like:



then it is reasonable to conclude that overweight and lack of exercise are highly correlated with diabetes.

Of course, this is an oversimplified example. Not all correlations are as evident. Health records actually include much more comprehensive information, such as age, gender, ethnicity, cholesterol levels, etc. This would produce data vectors $\mathbf{x}_i$ in higher dimensions:

$$\mathbf{x}_i = \begin{bmatrix} \text{weight} \\ \text{exercise} \\ \text{glucoseage} \\ \text{gender} \\ \text{ethnicity} \\ \text{cholesterol} \\ \vdots \end{bmatrix} \in \mathbb{R}^D.$$

Now you will have to use your imagination to decide how D-dimensional space looks like. In fact, it can be very challenging to visualize points in $\mathbb{R}^D$ (with $D > 3$, obviously). Luckily, using linear algebra we can find *lines*, *planes*, *curves*, etc. (similar to the gray curves depicted in the figure above, only in higher dimensions) that explain out data (just as the gray curves explain the correlations between weight, exercise and glucose).

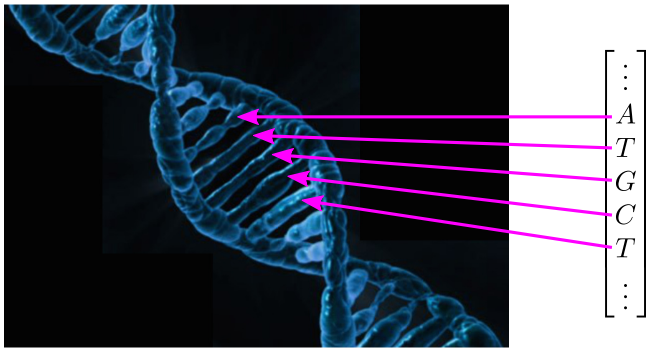
Example 4.3 (Recommender systems)**.** Similarly, Amazon, Netflix, Pandora, Spotify, Pinterest, Yelp, Apple, etc., keep information of their users, such as age, gender, income level, and very importantly, ratings of their products. The information of the i^{th} user can be arranged as a vector

$$\mathbf{x}_i = \begin{bmatrix} \text{age} \\ \text{gender} \\ \text{income} \\ \text{rating of item 1} \\ \text{rating of item 2} \\ \vdots \\ \text{rating of item } D-3 \end{bmatrix} \in \mathbb{R}^D.$$

In this sort of problem we want to analyze these data vectors to predict which users will like which items, in order to make good recommendations. If Amazon recommends you an item you will like, you are more likely to buy it. You can see why all these companies have a great interest in this problem, and they are paying *a lot* of money to people who work on this.

This can be done by finding structures (e.g., *lines* or *curves*) in high-dimensions that explain the data. As in Example 4.2, where we discovered that weight and exercise are good predictors for diabetes, here we want to discover which variables (e.g., gender, age, income, etc.) can predict which items (e.g., movies, shoes, songs, etc.) you would like.

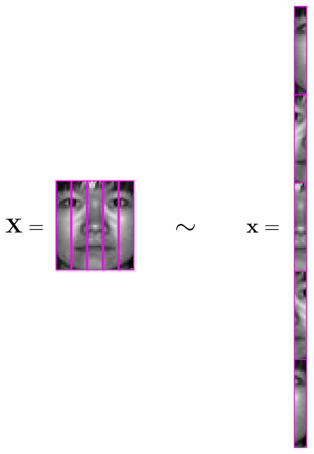
Example 4.4 (Genomics)**.** The genome of each individual can be stored as a vector containing its corresponding sequence of nucleotides, e.g., Adenine, Thymine, Guanine, Cytosine, Thymine, . . .



$$\begin{bmatrix} \vdots \\ A \\ T \\ G \\ C \\ T \\ \vdots \end{bmatrix} \sim \mathbf{x}_\mathrm{i} = \begin{bmatrix} \vdots \\ 1 \\ 4 \\ 3 \\ 2 \\ 4 \\ \vdots \end{bmatrix} \in \mathbb{R}^\mathrm{D}$$

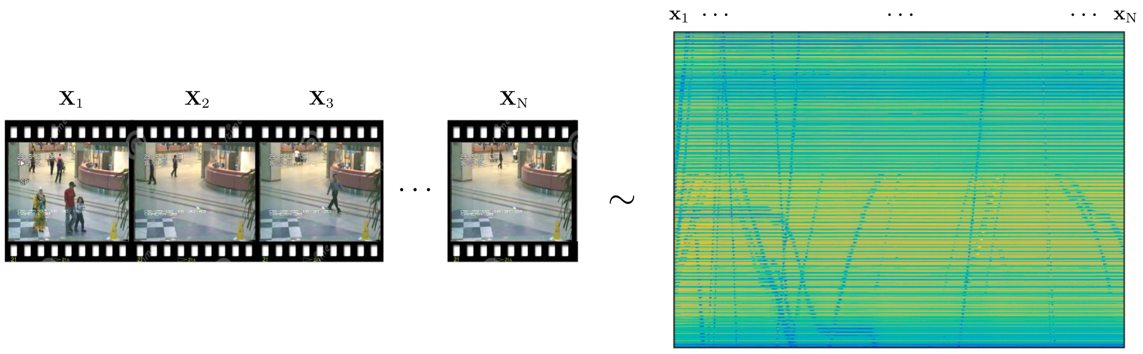
In this sort of problem we want to analyze these data vectors to determine which genes are correlated to which diseases (or features, like height or weight).

Example 4.5 (Image processing)**.** A $\mathrm{m} \times \mathrm{n}$ grayscale image can be stored in a data matrix $\mathbf{X} \in \mathbb{R}^{\mathrm{m}\times\mathrm{n}}$ whose $(\mathrm{i},\mathrm{j})^{\mathrm{th}}$ entry contains the gray intensity of pixel (i,j). Furthermore, $\mathbf{X}$ can be *vectorized*, i.e., we can stack its columns to form a vector $\mathbf{x} \in \mathbb{R}^\mathrm{D}$, with $\mathrm{D} = \mathrm{mn}$.



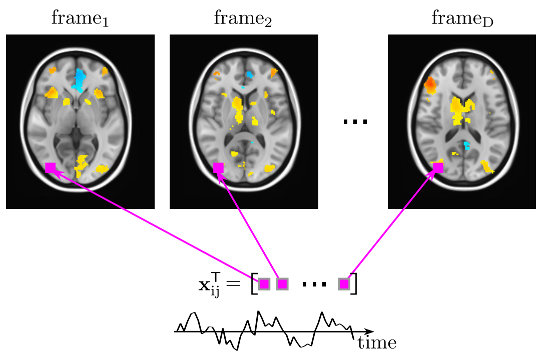
We want to analyze these vectors to interpret the image. For example, identify the objects that appear in the image, classifying faces, etc.

Example 4.6 (Computer vision)**.** The images $\mathbf{X}_1, \dots, \mathbf{X}_N \in \mathbb{R}^{m \times n}$ that form a video can be vectorized to obtain vectors $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^D$.



Similar to image processing, we want to analyze these vectors to interpret the video. For example, be able to distinguish background from foreground, track objects, etc. This has applications in surveillance, defense, robotics, etc.

Example 4.7 (Neural activity)**.** *Functional magnetic resonance imaging* (fMRI) generates a series of MRI images over time. Because oxygenated and deoxygenated hemoglobin have slightly different magnetic characteristics, variations in the MRI intensity indicate areas of the brain with increased blood flow and hence neural activity. The central task in fMRI is to reliably detect neural activity at different spatial locations (pixels) in the brain. The measurements over time at the $(\mathrm{i}, \mathrm{j})^{\mathrm{th}}$ pixel can be stored in a data vector $\mathbf{x}_{ij} \in \mathbb{R}^D$.

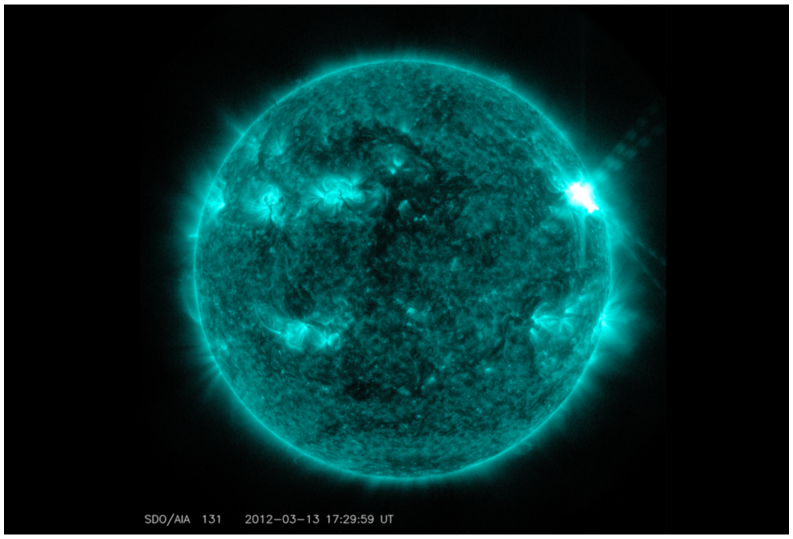


The idea is to analyze these vectors to determine the active pixels.

Example 4.8 (Sun flares)**.** The Sun, like all active stars, is constantly producing huge electromagnetic *flares*. Every now and then, these flares hit the Earth. Last time this happened was in 1859, and all that happened was that you could see the northern lights all the way down to Mexico — not a bad secondary effect! However, back in 1859 we didn't have a massive power grid, satellites, wireless communications, GPS, airplanes, space stations, etc. If a flare hits the Earth now, all these systems would be crippled, and repairing them could take *years* and would cost *trillions* of dollars to the U.S. alone! To make things worse, it turns out that these flares are not rare at all! It is estimated that the chance that a flare hits the earth in the next decade is about 12%.

Of course, we cannot stop these flares any more than we can stop an earthquake. If it hits us, it hits us. However, like with an earthquake, we can act ahead. If we know that one flare is coming, we can turn everything off, let it pass, and then turn everything back on, like nothing happened. Hence the NASA and other institutions are investing a great deal of time, effort and money to develop techniques that enable us to *predict* that a flare is coming.

So essentially, we want to device a sort of flares *radar* or *detector*. This radar would receive, for example, an image $\mathbf{X}$ of the sun (or equivalently, a vectorized image $\mathbf{x} \in \mathbb{R}^D$), and would have to decide whether a flare is coming or not.



These are only a few examples that I hope help convince you that vectors are the backbone of data science. In these notes we will review some of the most basic linear algebra concepts that will later enable us to use powerful vectors machinery that has been developed over centuries in order to tackle the modern problems in data science.

4.2 Fundamental Concepts

One of the most elemental vector manipulations are linear combinations, which essentially means scaling and adding vectors.

Definition 4.1 (Linear combination, coefficients)**.** A vector $\mathbf{z}$ is a *linear combination* of $\{\mathbf{x}_1, \ldots, \mathbf{x}_R\}$ if it can be written as

$$\mathbf{z} = \sum_{r=1}^{R} c_r \mathbf{x}_r \tag{4.1}$$

for some $c_1, \ldots, c_R \in \mathbb{R}$. The scalars $\{c_1, \ldots, c_R\}$ are called the *coefficients* of $\mathbf{z}$ with respect to (w.r.t.) $\{\mathbf{x}_1, \ldots, \mathbf{x}_R\}$.

Example 4.9. Let $\mathbf{x}$ and $\mathbf{y}$ be as in Example 4.1. Let

$$\mathbf{z} \;=\; -3\mathbf{x}+2\mathbf{y} \;=\; -3\begin{bmatrix}1\\-2\\\pi\end{bmatrix}+2\begin{bmatrix}e\\0\\1\end{bmatrix} \;=\; \begin{bmatrix}-3\\6\\-3\pi\end{bmatrix}+\begin{bmatrix}2e\\0\\2\end{bmatrix} \;=\; \begin{bmatrix}-3+2e\\6\\-3\pi+2\end{bmatrix}$$

Then $\mathbf{z}$ is a linear combination of $\mathbf{x}$ and $\mathbf{y}$, with coefficients -3 and 2.

Another fundamental concept is linear independence.

Definition 4.2 (Linear independence)**.** A set of vectors $\{\mathbf{x}_1,\dots,\mathbf{x}_\mathrm{R}\}$ is *linearly independent* if

$$\sum_{\mathrm{r}=1}^{\mathrm{R}} \mathrm{c}_\mathrm{r}\mathbf{x}_\mathrm{r} \;=\; \mathbf{0}$$

implies $\mathrm{c}_\mathrm{r}=0$ for every $\mathrm{r}=1,\dots,\mathrm{R}$. Otherwise we say it is *linearly dependent.*

Intuitively, a set of vectors is linearly independent if none of them can be written as a linear combination of the others.

Example 4.10. The following vectors are linearly independent:

$$\mathbf{x}_1 \;=\; \begin{bmatrix}1\\0\\0\end{bmatrix}, \qquad \mathbf{x}_2 \;=\; \begin{bmatrix}0\\1\\0\end{bmatrix}, \qquad \mathbf{x}_3 \;=\; \begin{bmatrix}0\\0\\1\end{bmatrix},$$

because we cannot write either of them as a linear combination of the others. In contrast, the following vectors are linearly dependent:

$$\mathbf{y}_1 \;=\; \begin{bmatrix}5\\5\\2\end{bmatrix}, \qquad \mathbf{y}_2 \;=\; \begin{bmatrix}1\\1\\0\end{bmatrix}, \qquad \mathbf{y}_3 \;=\; \begin{bmatrix}0\\0\\1\end{bmatrix},$$

because we can write $\mathbf{x}_1$ as a linear combination of $\mathbf{x}_2$ and $\mathbf{x}_3$, namely, $\mathbf{x}_1 = 5\mathbf{x}_2 + 2\mathbf{x}_3$.

4.3 Matrices

Matrices are very handy structures to arrange and manipulate vectors.

Example 4.11. We can arrange the vectors in the Example 4.10 in the following matrices:

$$\mathbf{X} \;=\; \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}, \qquad \mathbf{Y} \;=\; \begin{bmatrix}5&1&0\\5&1&0\\2&0&1\end{bmatrix}.$$

4.3.1 Review of Basic Matrix Operations

- **Matrix Multiplication.** Given matrices $\mathbf{A} \in \mathbb{R}^{m \times k}$ and $\mathbf{B} \in \mathbb{R}^{k \times n}$, their product, denoted by $\mathbf{AB}$, is another matrix of size $m \times n$ whose $(i,j)^{th}$ entry is given by:

$$[\mathbf{AB}]_{ij} = \sum_{\ell=1}^{k} \mathbf{A}_{i\ell}\mathbf{B}_{\ell j}.$$

  Intuitively, the $(i,j)^{th}$ entry of $\mathbf{AB}$ is given by the multiplication of the i^{th} row of $\mathbf{A}$ and the j^{th} column of $\mathbf{B}$. For example, if

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}, \tag{4.2}$$

  Then:

$$\mathbf{AB} = \begin{bmatrix} (1\cdot 1 + 2\cdot 3 + 3\cdot 5) & (1\cdot 2 + 2\cdot 4 + 3\cdot 6) \\ (4\cdot 1 + 5\cdot 3 + 6\cdot 5) & (4\cdot 2 + 5\cdot 4 + 6\cdot 6) \end{bmatrix} = \begin{bmatrix} 22 & 28 \\ 49 & 64 \end{bmatrix}.$$

- **Scalar Multiplication** Given a scalar c and a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, their product, denoted by $c\mathbf{A}$, is an $m \times n$ matrix whose $(i,j)^{th}$ entry is given by c times the $(i,j)^{th}$ entry of $\mathbf{A}$. For example, with $\mathbf{A}$ as in (4.2), and $c = 7$,

$$c\mathbf{A} = 7\mathbf{A} = \begin{bmatrix} 7 & 14 & 21 \\ 28 & 35 & 42 \end{bmatrix},$$

- **Transposition.** The transpose of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, denoted by $\mathbf{A}^\mathsf{T}$, is an $n \times m$ matrix whose $(i,j)^{th}$ entry is given by the $(j,i)^{th}$ entry of $\mathbf{A}$. Intuitively, transposing a matrix is like flipping its rows and columns along the diagonal. For example, with $\mathbf{A}$ as in (4.2),

$$\mathbf{A}^\mathsf{T} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}.$$

- **Trace.** Given a squared matrix $\mathbf{X} \in \mathbb{R}^{m \times m}$, its trace, denoted by $\mathsf{tr}(\mathbf{X})$ is the sum of its diagonal entries:

$$\mathsf{tr}(\mathbf{X}) = \sum_{i=1}^{m} \mathbf{X}_{ii}.$$

  For example, with $\mathbf{X}$ as in Example 4.11, $\mathsf{tr}(\mathbf{X}) = 3$.

- **Identity Matrix.** The identity matrix of size $m \times m$, denoted by $\mathbf{I}_m$, is a squared matrix whose diagonal entries are all ones, and off-diagonal entries are all zeros. For example, the 3×3 identity matrix is

$$\mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

  Whenever there is no possible confusion about the size of the identity matrix, people often drop the m subindex, and simply denote it as $\mathbf{I}$.

- **Inverse.** Given a squared matrix $\mathbf{X} \in \mathbb{R}^{\mathrm{m}\times\mathrm{m}}$, its inverse, denoted by $\mathbf{X}^{-1}$ is a matrix such that $\mathbf{X}\mathbf{X}^{-1} = \mathbf{X}^{-1}\mathbf{X} = \mathbf{I}$. For example, with $\mathbf{X}$ as in Example 4.11, $\mathbf{X}^{-1} = \mathbf{X}$. As another example, consider

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix}.$$

Then

$$\mathbf{X}^{-1} = \begin{bmatrix} 3 & -3 & 1 \\ -2.5 & 4 & -1.5 \\ 0.5 & -1 & 0.5 \end{bmatrix}.$$

You can verify that $\mathbf{X}\mathbf{X}^{-1} = \mathbf{X}^{-1}\mathbf{X} = \mathbf{I}$. Here is a special trick to invert 2×2 matrices:

$$\begin{bmatrix} \mathrm{a} & \mathrm{b} \\ \mathrm{c} & \mathrm{d} \end{bmatrix}^{-1} = \frac{1}{\mathrm{ad}-\mathrm{bc}} \begin{bmatrix} \mathrm{d} & -\mathrm{b} \\ -\mathrm{c} & \mathrm{a} \end{bmatrix}.$$

Of course, this requires that $\mathrm{ad} - \mathrm{bc} \neq 0$.

- **Hadamard Product.** Given two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{\mathrm{m}\times\mathrm{n}}$, their Hadamard product (aka point-wise product) is a matrix of size $\mathrm{m} \times \mathrm{n}$, denoted as $\mathbf{A} \odot \mathbf{B}$, whose $(\mathrm{i},\mathrm{j})^{\mathrm{th}}$ entry is given by the product of the (i,j) entries of $\mathbf{A}$ and $\mathbf{B}$, i.e.,

$$[\mathbf{A} \odot \mathbf{B}]_{\mathrm{ij}} = \mathbf{A}_{\mathrm{ij}}\mathbf{B}_{\mathrm{ij}}.$$

For example, with $\mathbf{A}, \mathbf{B}$ as in (4.2),

$$\mathbf{A} \odot \mathbf{B}^{\mathsf{T}} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \odot \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 6 & 15 \\ 8 & 20 & 36 \end{bmatrix}.$$

- **Vector Operations.** Notice that vectors are 1-column matrices, so all matrix operators that apply to non-squared matrices also apply to vectors. For instance, with the same setup as in Example 4.10,

$$\mathbf{x}_1^{\mathsf{T}}\mathbf{x}_2 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = 0.$$

Notice that $\mathbf{x}_1^{\mathsf{T}}\mathbf{x}_2 \neq \mathbf{x}_1\mathbf{x}_2^{\mathsf{T}}$:

$$\mathbf{x}_1\mathbf{x}_2^{\mathsf{T}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

- **Norms.** Given a matrix $\mathbf{A} \in \mathbb{R}^{\mathrm{m}\times\mathrm{n}}$, its Frobenius norm is defined as:

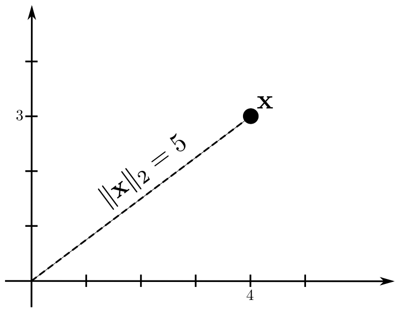
$$\|\mathbf{A}\|_{\mathrm{F}} := \sqrt{\sum_{\mathrm{i}=1}^{\mathrm{m}}\sum_{\mathrm{j}=1}^{\mathrm{n}} \mathbf{A}_{\mathrm{ij}}^2}.$$

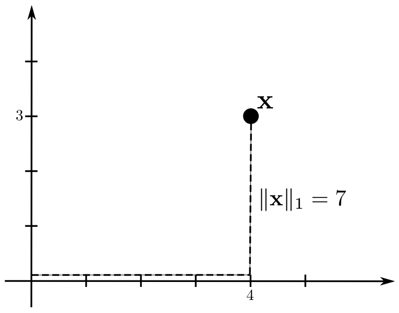
In the particular case of vectors, this is often called the ℓ_2-norm or Euclidean norm, and is denoted by $\|\mathbf{x}\|_2$, or simply $\|\mathbf{x}\|$. Norms are important because they essentially quantify the *size* of a matrix or vector. Just as there are several ways to quantify the size of a person (e.g., age, height, weight), there are also several ways to quantify the *size* of a matrix or vector, for which we can use different norms. Another example is the ℓ_1-norm (aka taxi-cab or Manhattan norm) is defined as:

$$\|\mathbf{A}\|_1 := \sum_{i=1}^{m}\sum_{j=1}^{n} |\mathbf{A}_{ij}|.$$

Intuitively, the ℓ_2-norm measures the *point-to-point* size, while the ℓ_1-norm measures the *taxi-cab* size. For example, for the same vector $\mathbf{x} = [4 \;\; 3]^\mathsf{T}$, here are two different notions to quantify its size:

$$\|\mathbf{x}\|_2 = \sqrt{4^2+3^2} = 5 \qquad\qquad \|\mathbf{x}\|_1 = |4| + |3| = 7.$$





The norm $\|\mathbf{x}\|$ of a vector $\mathbf{x}$ is essentially its size. Norms are also useful because they allow us to measure distance between vectors (through their difference). For example, consider the following images:

and vectorize them as in Example 4.5 to produce vectors $\mathbf{x}, \mathbf{y}, \mathbf{z}$. We want to do face clustering, i.e., we want to know which images correspond to the same person. If $\|\mathbf{x} - \mathbf{y}\|$ is small (i.e., $\mathbf{x}$ is similar to $\mathbf{y}$), it is reasonable to conclude that the first two images correspond to the same person. If $\|\mathbf{x} - \mathbf{z}\|$ is large (i.e., $\mathbf{x}$ is very different from $\mathbf{z}$), it is reasonable to conclude that the first and second images corresponds to different persons.

Norms satisfy the so-called *triangle inequality*: $\|\mathbf{x}+\mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$. This allows you to draw useful conclusions. For example, knowing that $\|\mathbf{x}-\mathbf{y}\|$ is small and that $\|\mathbf{x}-\mathbf{z}\|$ is large allows us to conclude that $\|\mathbf{y} - \mathbf{z}\|$ is also large. Intuitively, this allows us to conclude that if $\mathbf{x}, \mathbf{y}$ correspond to the same person, and $\mathbf{x}$ and $\mathbf{z}$ corresponds to different persons, then $\mathbf{y}$ and $\mathbf{z}$ also correspond to different persons. In other words, nothing weird will happen.

Inner Product. Given two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$, their inner product is defined as:

$$\langle \mathbf{A}, \mathbf{B} \rangle := \mathsf{tr}(\mathbf{A}^\mathsf{T}\mathbf{B}).$$

For example, with $\mathbf{A}$ as in (4.2),

$$\langle \mathbf{A}, \mathbf{A} \rangle = \mathsf{tr}(\mathbf{A}^\mathsf{T}\mathbf{A}) = \mathsf{tr}\left(\begin{bmatrix}1 & 4\\2 & 5\\3 & 6\end{bmatrix}\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\end{bmatrix}\right) = \mathsf{tr}\left(\begin{bmatrix}17 & 22 & 27\\22 & 29 & 36\\27 & 36 & 45\end{bmatrix}\right) = 91.$$

Notice that for vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{\mathrm{D}}$, $\mathbf{x}^{\mathsf{T}}\mathbf{y}$ will always be a scalar, so we can drop the trace, and simply write $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^{\mathsf{T}}\mathbf{y}$. For example, with the same setup as in Example 4.10:

$$\begin{aligned} \langle \mathbf{x}_1, \mathbf{x}_2 \rangle &= \mathbf{x}_1^{\mathsf{T}}\mathbf{x}_2 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = 0, \\ \langle \mathbf{y}_1, \mathbf{y}_2 \rangle &= \mathbf{y}_1^{\mathsf{T}}\mathbf{y}_2 = \begin{bmatrix} 5 & 5 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = 10. \end{aligned}$$

Inner products are of particular importance because they measure the similarity between matrices and vectors. In particular, recall from your kinder garden classes that the angle θ between two vectors is given by:

$$\cos\theta = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\|\|\mathbf{y}\|}.$$

More generally, the larger the inner product between two objects (in absolute value), the more similar they are.

4.3.2 Why do I care about Matrices?

Matrix operators are useful because they allow us to write otherwise complex burdensome operations in simple and concise matrix form.

Example 4.12. With the same settings as in Examples 4.10 and 4.11, we can write linear combinations using a simple matrix multiplication. For instance, instead of writing

$$\mathbf{y}_1 = \begin{bmatrix} 5 \\ 5 \\ 2 \end{bmatrix} = 5 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + 5 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 5\mathbf{x}_1 + 5\mathbf{x}_2 + 2\mathbf{x}_3,$$

we can let $\mathbf{c} = [5 \;\; 5 \;\; 2]^{\mathsf{T}}$ be $\mathbf{y}$'s coefficient vector, and equivalently write in simpler form:

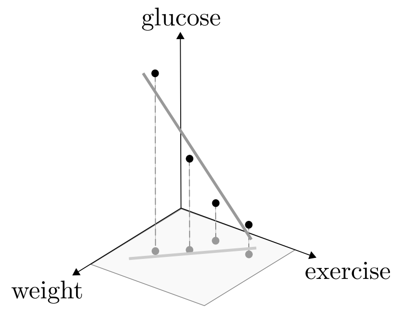
$$\mathbf{y}_1 = \begin{bmatrix} 5 \\ 5 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 5 \\ 2 \end{bmatrix} = \mathbf{X}\mathbf{c}.$$

Similar matrix expressions will be ubiquitous throughout this course, so you should start getting familiar and feeling comfortable using matrix operations.

4.4 Subspaces

Subspaces are essentially high-dimensional lines. A 1-dimensional subspace is a line, a 2-dimensional subspaces is a plane, and so on. Subspaces are useful because data often lies near subspaces.

Example 4.13. The health records data in Example 4.2 lies near a 1-dimensional subspace (line):



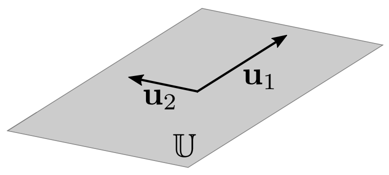
In higher dimensions subspaces may be harder to visualize, so you will have to use imagination to decide how a higher-dimensional subspace looks. Luckily, we have a precise and formal mathematical way to define them:

Definition 4.3 (Subspace)**.** A subset $\mathbb{U} \subseteq \mathbb{R}^D$ is a *subspace* if for every $a, b \in \mathbb{R}$ and every $\mathbf{u}, \mathbf{v} \in \mathbb{U}$, $a\mathbf{u} + b\mathbf{v} \in \mathbb{U}$.

Definition 4.4 (Span)**.** $\text{span}[\mathbf{u}_1, \ldots, \mathbf{u}_R]$ is the set of all linear combinations of $\{\mathbf{u}_1, \ldots, \mathbf{u}_R\}$. More formally,

$$\text{span}[\mathbf{u}_1, \ldots, \mathbf{u}_R] \ := \ \left\{ \mathbf{x} \in \mathbb{R}^D \ : \ \mathbf{x} = \sum_{r=1}^{R} c_r \mathbf{u}_r \ \text{ for some } c_1, \ldots, c_R \in \mathbb{R} \right\}.$$

For any vectors $\mathbf{u}_1, \ldots, \mathbf{u}_R \in \mathbb{R}^D$, $\text{span}[\mathbf{u}_1, \ldots, \mathbf{u}_R]$ is a subspace. For example, here is a subspace $\mathbb{U}$ (plane) spanned by two vectors, $\mathbf{u}_1$ and $\mathbf{u}_2$:



Definition 4.5 (Basis)**.** A set of linearly independent vectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_R\}$ is a *basis* of a subspace $\mathbb{U}$ if each $\mathbf{v} \in \mathbb{U}$ can be written as

$$\mathbf{v} \ = \ \sum_{r=1}^{R} c_r \mathbf{u}_r$$

for a *unique* set of coefficients $\{c_1, \ldots, c_R\}$.

Definition 4.6 (Orthogonal)**.** A collection of vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_\mathrm{R}\}$ is *orthogonal* if $\langle \mathbf{x}_\mathrm{i}, \mathbf{x}_\mathrm{j} \rangle = 0$ for every $\mathrm{i} \neq \mathrm{j}$.

If $\|\mathbf{x}\| = 1$, we say $\mathbf{x}$ is a *unit* vector, or that it is *normalized.* Similarly, a collection of normalized, orthogonal vectors is called *orthonormal.* There is a tight relation between inner products and norms. The following is one of the most important and useful inequalities that describe this relationship.

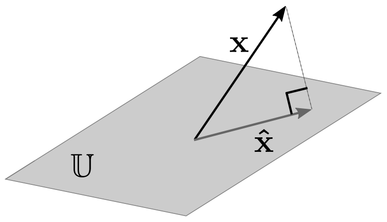
Proposition 4.1 (Cauchy-Schwartz inequality)**.** For every $\mathbf{x}, \mathbf{y} \in \mathbb{R}^\mathrm{D}$,

$$|\langle \mathbf{x}, \mathbf{y} \rangle| \;\leq\; \|\mathbf{x}\| \|\mathbf{y}\|.$$

Furthermore, if $\mathbf{y} \neq \mathbf{0}$, then equality holds if and only if $\mathbf{x} = \mathrm{a}\mathbf{y}$ for some $\mathrm{a} \in \mathbb{R}$.

4.5 Projections

In words, the projection $\hat{\mathbf{x}}$ of a vector $\mathbf{x}$ onto a subspace $\mathbb{U}$ is the vector in $\mathbb{U}$ that is closest to $\mathbf{x}$:



More formally,

Definition 4.7 (Projection)**.** The *projection* of $\mathbf{x} \in \mathbb{R}^\mathrm{D}$ onto subspace $\mathbb{U}$ is the vector $\hat{\mathbf{x}} \in \mathbb{U}$ satisfying

$$\|\mathbf{x} - \hat{\mathbf{x}}\| \;\leq\; \|\mathbf{x} - \mathbf{u}\| \qquad \text{for every } \mathbf{u} \in \mathbb{U}.$$

Notice that if $\mathbf{x} \in \mathbb{U}$, then $\hat{\mathbf{x}} = \mathbf{x}$. The following proposition tells us exactly how to compute projections.

Proposition 4.2. Let $\{\mathbf{u}_1, \ldots, \mathbf{u}_\mathrm{R}\}$ be an orthonormal basis of $\mathbb{U}$. The projection of $\mathbf{x} \in \mathbb{R}^\mathrm{D}$ onto $\mathbb{U}$ is given by

$$\hat{\mathbf{x}} \;=\; \sum_{\mathrm{r}=1}^{\mathrm{R}} \langle \mathbf{x}, \mathbf{u}_\mathrm{r} \rangle \mathbf{u}_\mathrm{r}.$$

In other words, the coefficient of $\mathbf{x}$ w.r.t. $\mathbf{u}_\mathrm{r}$ is given by $\langle \mathbf{x}, \mathbf{u}_\mathrm{r} \rangle$.

Furthermore, the following proposition tells us that we can compute projections very efficiently: just using a simple matrix multiplication! This makes projections very attractive in practice. For example, as we saw before, data often lies near subspaces. We can measure how close using the norm of the *residual* $\mathbf{x} - \hat{\mathbf{x}}$.

Proposition 4.3 (Projector operator)**.** Let $\mathbf{U} \in \mathbb{R}^{\mathrm{D}\times\mathrm{R}}$ be a basis of $\mathbb{U}$. The *projection operator* $\mathbf{P_U} : \mathbb{R}^{\mathrm{D}} \to \mathbb{U}$ that maps any vector $\mathbf{x} \in \mathbb{R}^{\mathrm{D}}$ to its projection $\hat{\mathbf{x}} \in \mathbb{U}$ is given by:

$$\mathbf{P_U} \;=\; \mathbf{U}(\mathbf{U}^{\mathsf{T}}\mathbf{U})^{-1}\mathbf{U}^{\mathsf{T}}.$$

Notice that if $\mathbf{U}$ is orthonormal, then $\mathbf{P_U} = \mathbf{U}\mathbf{U}^{\mathsf{T}}$.

Proof. Since $\hat{\mathbf{x}} \in \mathbb{U}$, that means we can write $\hat{\mathbf{x}}$ as $\mathbf{Uc}$ for some $\mathbf{c} \in \mathbb{R}^{\mathrm{R}}$. We thus want to find the $\mathbf{c}$ that minimizes:

$$\|\mathbf{x} - \mathbf{Uc}\|_2^2 \;=\; (\mathbf{x} - \mathbf{Uc})^{\mathsf{T}}(\mathbf{x} - \mathbf{Uc}) \;=\; \mathbf{x}^{\mathsf{T}} - 2\mathbf{c}^{\mathsf{T}}\mathbf{U}^{\mathsf{T}}\mathbf{x} + \mathbf{c}^{\mathsf{T}}\mathbf{U}^{\mathsf{T}}\mathbf{Uc}.$$

Since this is convex in $\mathbf{c}$, we can use elemental optimization to find the desired minimizer, i.e., we will take derivative w.r.t. $\mathbf{c}$, set to zero and solve for $\mathbf{c}$. To learn more about how to take derivatives w.r.t. vectors and matrices see *Old and new matrix algebra useful for statistics* by Thomas P. Minka. The derivative w.r.t. $\mathbf{c}$ is given by:

$$-2\mathbf{U}^{\mathsf{T}}\mathbf{x} + 2\mathbf{U}^{\mathsf{T}}\mathbf{Uc}.$$

Setting to zero and solving for $\mathbf{c}$ we obtain:

$$\hat{\mathbf{c}} \;:=\; \operatorname*{arg\,min}_{\mathbf{c}\in\mathbb{R}^{\mathrm{R}}} \;\|\mathbf{x} - \mathbf{Uc}\|_2^2 \;=\; (\mathbf{U}^{\mathsf{T}}\mathbf{U})^{-1}\mathbf{U}^{\mathsf{T}}\mathbf{x},$$

where we know $\mathbf{U}^{\mathsf{T}}\mathbf{U}$ is invertible because $\mathbf{U}$ is a basis by assumption, so its columns are linearly independent. It follows that

$$\hat{\mathbf{x}} \;=\; \mathbf{U}\hat{\mathbf{c}} \;=\; \underbrace{\mathbf{U}(\mathbf{U}^{\mathsf{T}}\mathbf{U})^{-1}\mathbf{U}^{\mathsf{T}}}_{\mathbf{P_U}}\mathbf{x},$$

as claimed. If $\mathbf{U}$ is orthonormal, then $\mathbf{U}^{\mathsf{T}}\mathbf{U} = \mathbf{I}$, and hence $\mathbf{P_U}$ simplifies to $\mathbf{U}\mathbf{U}^{\mathsf{T}}$. Notice that $\hat{\mathbf{c}}$ are the coefficients of $\hat{\mathbf{x}}$ w.r.t. the basis $\mathbf{U}$. □

4.6 Gram-Schmidt Orthogonalization

Orthonormal bases have very nice and useful properties. For example, in Proposition 4.3, if the basis $\mathbf{U}$ is orthonormal, then the projection operator is simplified into $\mathbf{U}\mathbf{U}^{\mathsf{T}}$, which requires much less computations than $\mathbf{U}(\mathbf{U}^{\mathsf{T}}\mathbf{U})^{-1}\mathbf{U}^{\mathsf{T}}$. The following procedure tells us exactly how to transform an arbitrary basis into an orthonormal basis.

Proposition 4.4 (Gram-Schmidt procedure)**.** Let $\{\mathbf{u}_1, \dots, \mathbf{u}_\mathrm{R}\}$ be a basis of $\mathbb{U}$. Let

$$\begin{aligned} \mathbf{v}'_\mathrm{r} &= \begin{cases} \mathbf{u}_1 & \mathrm{r} = 1, \\ \mathbf{u}_\mathrm{r} - \sum_{\mathrm{k}=1}^{\mathrm{r}-1} \langle \mathbf{u}_\mathrm{r}, \mathbf{v}_\mathrm{k} \rangle \mathbf{v}_\mathrm{k} & \mathrm{r} = 2, \dots, \mathrm{R}, \end{cases} \\ \mathbf{v}_\mathrm{r} &= \mathbf{v}'_\mathrm{r} / \| \mathbf{v}'_\mathrm{r} \|. \end{aligned}$$

Then $\{\mathbf{v}_1, \dots, \mathbf{v}_\mathrm{R}\}$ are an orthonormal basis of $\mathbb{U}$.

Proof. We know from Proposition 4.2 that $\sum_{\mathrm{k}=1}^{\mathrm{r}-1} \langle \mathbf{u}_\mathrm{r}, \mathbf{v}_\mathrm{k} \rangle \mathbf{v}_\mathrm{k}$ is the projection of $\mathbf{u}_\mathrm{r}$ onto span$[\mathbf{v}_1, \dots, \mathbf{v}_{\mathrm{r}-1}]$. This implies $\mathbf{v}'_\mathrm{r}$ is the orthogonal residual of $\mathbf{u}_\mathrm{r}$ onto span$[\mathbf{v}_1, \dots, \mathbf{v}_{\mathrm{r}-1}]$, and hence it is orthogonal to $\{\mathbf{v}_1, \dots, \mathbf{v}_{\mathrm{r}-1}\}$, as desired. $\mathbf{v}_\mathrm{r}$ is simply the normalized version of $\mathbf{v}'_\mathrm{r}$. $\square$

4.7 Conclusions

These notes review several fundamental concepts of linear algebra that lie at the heart of data science, and will be crucial in *every* topic to be studied in this course.