**CS 4980/6980: Introduction to Data Science** © Spring 2018

# Lecture 1: Introduction to Data Science

Instructor: Daniel L. Pimentel-Alarcón Scribed by: Daniel L. Pimentel-Alarcón

## 1.1 Introduction

In a nutshell, data science is all about obtaining *insights* from data. For example:

- Predict stock prices.
- Identify who may develop a disease (diabetes, cancer, Alzheimer, etc.).
- Identify which genes trigger a certain disease.
- Determine how species evolve, in particular bacteria and viruses that may be harmful to people.
- Classify things. For example, people, using faces or walk patterns.
- Understand how the brain works. For example, analyzing brain images/videos/data.
- Predict solar flares.
- Identify fraud (credit, clicks, identity, etc.).
- Targeted advertisement (recommender systems).

Modern data science requires mostly a combination of

- **Mathematical and statistical knowledge.** To know *how* to analyze data and draw sensible/accurate/correct conclusions.
- **Computer science skills.** To be able to use computers to analyze data in efficient ways.

However, data science is a highly interdisciplinary field. One can find data scientists with backgrounds in engineering, biology, chemistry, medicine, and virtually every field of knowledge.

## 1.2 An Intuitive Example

Consider the following data matrix containing health related information:

| **Variables** | Individual 1 | Individual 2 | Individual 3 | $\cdots$ | Individual N |
|---|---|---|---|---|---|
| Height | 5'10" | 5'7" | 6'1" | $\cdots$ | 5'5" |
| Weight | 150 | 200 | 180 | $\cdots$ | 145 |
| Glucose | 145 | 195 | 150 | $\cdots$ | 205 |

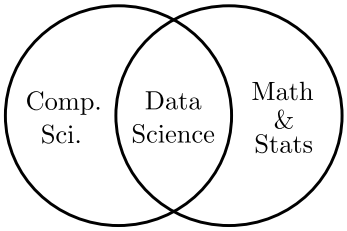



Figure 1.1: Modern data science mostly requires computer science, mathematics and statistics.

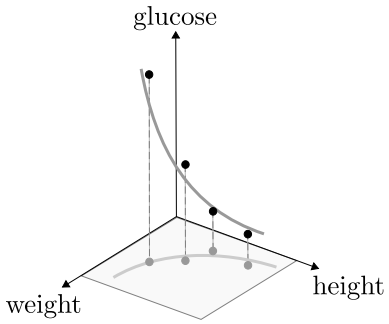
| | Examples | Regular | Bold | Lower | Capital | Roman | Script |
|---|---|---|---|---|---|---|---|
| Scalar | x | ✓ | | ✓ | | ✓ | |
| [Column] vector | $\mathbf{x}$ | | ✓ | ✓ | | ✓ | |
| Matrix | $\mathbf{X}$ | | ✓ | | ✓ | ✓ | |
| Random variable | $x$ | ✓ | | ✓ | | | ✓ |
| Random vector | $\boldsymbol{x}$ | | ✓ | ✓ | | | ✓ |
| Random matrix | $\boldsymbol{X}$ | | ✓ | | ✓ | | ✓ |

Table 1.1: Throughout this course we will use standard mathematical notation.

Variables (height, weight, etc.) are often also called *features*. This way, each individual is represented by its *features vector*:

$$\mathbf{x}_1 = \begin{bmatrix} 5'10" \\ 150 \\ 145 \end{bmatrix} \quad \mathbf{x}_2 = \begin{bmatrix} 5'7" \\ 200 \\ 195 \end{bmatrix} \quad \mathbf{x}_3 = \begin{bmatrix} 6'1" \\ 180 \\ 150 \end{bmatrix} \quad \cdots \quad \mathbf{x}_N = \begin{bmatrix} 5'5" \\ 145 \\ 205 \end{bmatrix}$$

Here each $\mathbf{x}_i$ is a [column] vector in $\mathbb{R}^3$. That is, a *point* in 3-dimensional space:



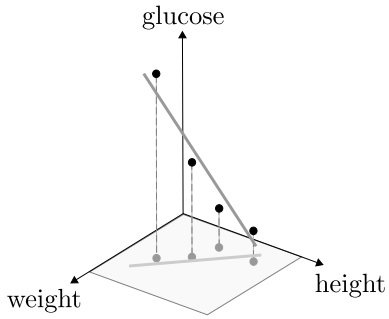

Feature vectors are also called *data points* (or simply *points*) or *training samples* (or simply *samples*).

With datasets like this, we may aim at several goals:

(a) Identify causes for diabetes. For example, we could learn that if the ratio weight/height is high, then individuals are more likely to develop diabetes.

(b) Predict who will develop diabetes. Given a new individual $\mathbf{x}_{N+1}$, what is the likelihood (probability) that she develops diabetes?

(c) Model the dependency of the variables of interest. For example, maybe we can approximate the variables' relationship as linear:



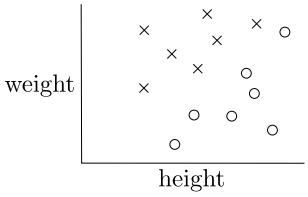

In (a) we already learnt that individuals with high weight/height ratio are more likely to develop diabetes. Models allow to estimate precisely *how likely* it is that an individual develops diabetes.

## 1.3 Categorical Data

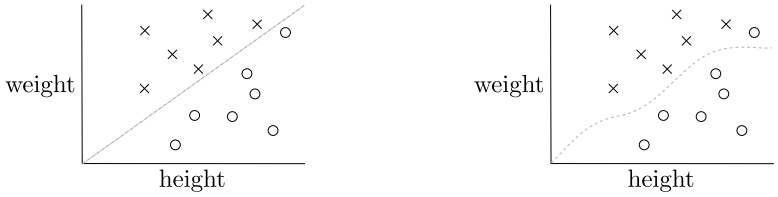
Notice that the three variables in $\mathbf{x}_i$ are *real-valued*. Sometimes we also have *categorical* variables. For example, rather than a glucose level, we may want to know whether an individual is healthy or diabetic:

| **Variables** | Individual 1 | Individual 2 | Individual 3 | $\cdots$ | Individual N |
|---|---|---|---|---|---|
| Height | 5'10" | 5'7" | 6'1" | $\cdots$ | 5'5" |
| Weight | 150 | 200 | 180 | $\cdots$ | 145 |
| Healthy/Diabetic | 0 | 1 | 0 | $\cdots$ | 1 |

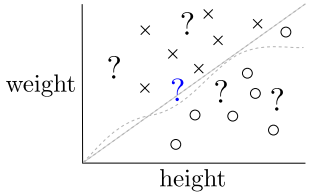
In such cases, we could visualize data as follows, where each circle represents a healthy individual, and each cross represents a diabetic individual:




We could then try to find a model that classifies this dataset, for example a *line* or a *polynomial*:




This would allow to classify new points (depicted as question marks) depending on their position in *feature space*:




Notice that depending on the model (classifier), the same point could be classified in different ways (see blue question mark). This raises the question of which model is better, which will later lead us to the concept of *overfitting*.

## 1.4 Multiple Categories

More generally, we could have several categories, rather than just two. For example, if instead of diabetes we were studying cancer, we could have a variable indicating the stage of a tumor (e.g., a value in $\{0, 1, 2, 3, 4\}$), rather than a variable in $\{0, 1\}$ indicating just healthy or diabetic.

| **Variables** | Individual 1 | Individual 2 | Individual 3 | $\cdots$ | Individual N |
|---|---|---|---|---|---|
| Height | 5'10" | 5'7" | 6'1" | $\cdots$ | 5'5" |
| Weight | 150 | 200 | 180 | $\cdots$ | 145 |
| Tumor stage | 2 | 0 | 4 | $\cdots$ | 1 |

How would you visualize data like this?

## 1.5 High-Dimensional Data

Of course, in general, we may have more than three variables:

| **Variables** | Individual 1 | Individual 2 | Individual 3 | $\cdots$ | Individual N |
|---|---|---|---|---|---|
| Height | 5'10" | 5'7" | 6'1" | $\cdots$ | 5'5" |
| Weight | 150 | 200 | 180 | $\cdots$ | 145 |
| Age | 30 | 50 | 25 | $\cdots$ | 40 |
| Gender | M | F | M | $\cdots$ | F |
| Glucose | 145 | 195 | 150 | $\cdots$ | 205 |
| Cholesterol | 110 | 140 | 100 | $\cdots$ | 160 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ |
| Tumor stage | 2 | 0 | 4 | $\cdots$ | 1 |

If we have D variables, then $\mathbf{x}_i \in \mathbb{R}^D$. How would you visualize data like this? How would you analyze it? Can you tell which variables determine cancer? Can you determine a model that describes the dependency between all variables?

These are the type of the questions that data scientists often want to answer.

## 1.6 Other Motivating Applications

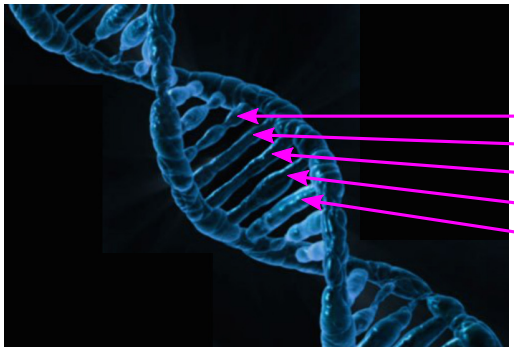
**Example 1.1** (Recommender systems)**.** Amazon, Netflix, Pandora, Spotify, Pinterest, Yelp, Apple, etc., keep information of their users, such as age, gender, income level, and very importantly, ratings of their products. The information of the $i^{th}$ user can be arranged as a vector:

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

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. In the example of Section 1.2 we discovered that the height/weight ratio is a good predictor 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.

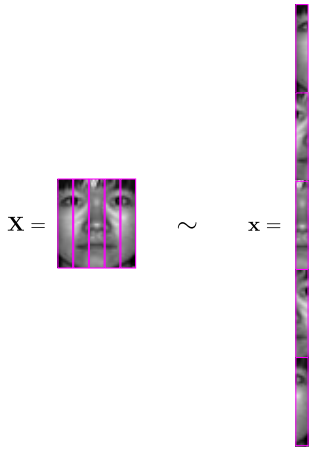
**Example 1.2** (Genomics)**.** The genome of each individual can be stored as a vector containing its corresponding sequence of nucleotides, e.g., Adenine, Thymine, Guanine, Cytosine, Thymine, ...



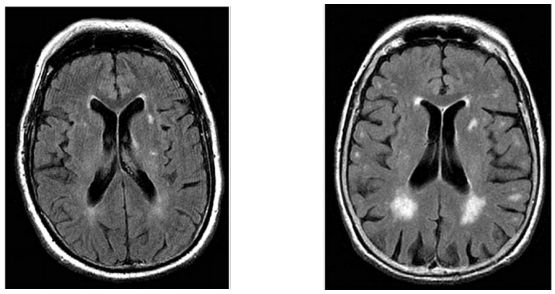
$$\begin{bmatrix} \vdots \\ A \\ T \\ G \\ C \\ T \\ \vdots \end{bmatrix} \sim \mathbf{x}_i = \begin{bmatrix} \vdots \\ 1 \\ 4 \\ 3 \\ 2 \\ 4 \\ \vdots \end{bmatrix} \in \mathbb{R}^D$$

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).

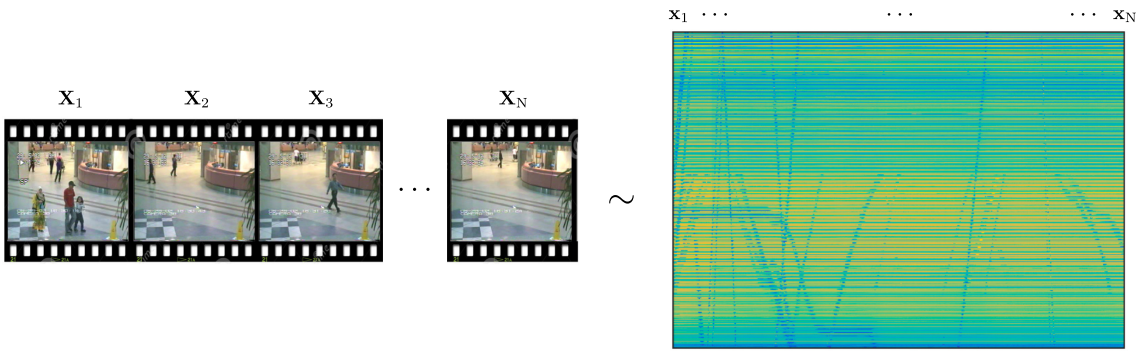
**Example 1.3** (Image processing)**.** A $m \times n$ grayscale image can be stored in a data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ whose $(i, j)^{th}$ entry contains the gray intensity of pixel $(i, j)$. Furthermore, $\mathbf{X}$ can be *vectorized*, i.e., we can stack its columns to form a vector $\mathbf{x} \in \mathbb{R}^D$, with $D = mn$.




We want to analyze these vectors to interpret the image. For example, identify the objects that appear in the image, classify faces, or even for medical diagnosis. For instance, can you tell which of these magnetic resonance images (MRIs) corresponds to an individual with Alzheimer?



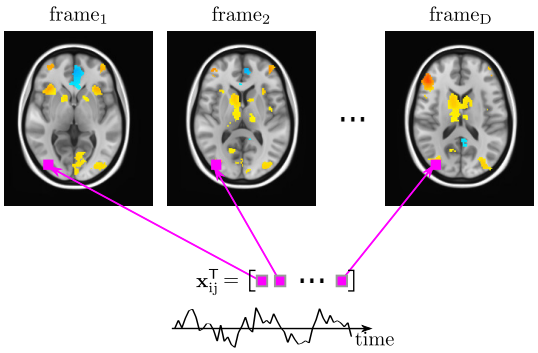
**Example 1.4** (Computer vision)**.** The images $\mathbf{X}_1, \ldots, \mathbf{X}_N \in \mathbb{R}^{m \times n}$ that form a video can be vectorized to obtain vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^D$.




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 1.5** (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 $(i, j)^{th}$ pixel can be stored in a data vector $\mathbf{x}_{ij} \in \mathbb{R}^D$.

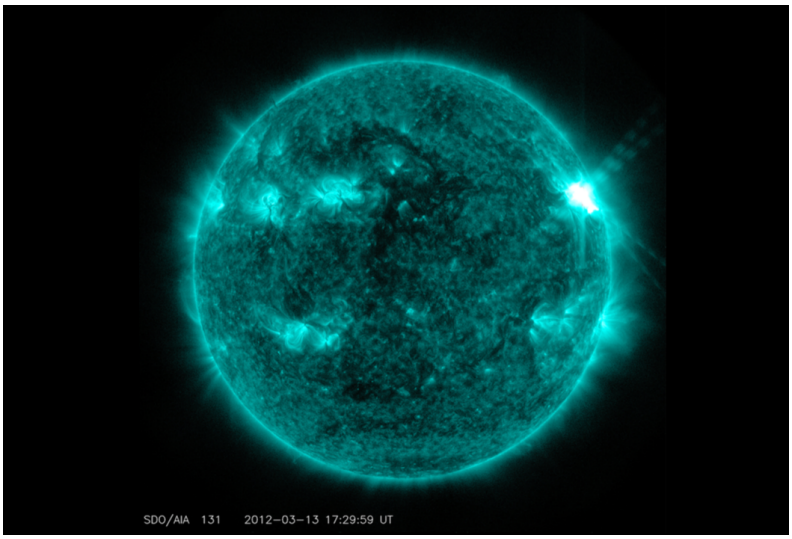



The idea is to analyze these vectors to determine the active pixels. This can help better understand how the brain works.

**Example 1.6** (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.




**Example 1.7** (Fraud detection)**.** Credit cards are a classical example of fraud detection. The main idea is to look at usage patterns, and identify *outliers* (unusual activity).

For a more interesting example, consider *click fraud*. Companies pay popular websites to advertise their products. How much they pay depends on the popularity of each website, measured in number of clicks. Hence, companies often cheat, using *bots* that click their websites (to have a higher click count, and charge more for advertising). How would you detect whether a click is genuine or fraudulent?

## 1.7 Conclusions

This lecture shows some motivating applications of data science. In all these applications, the task can be summarized as obtaining insights from data. This will often involve pre-processing (for example transforming real-valued glucose levels to a binary *label* indicating healthy/diabetic), visualization, analysis, classification, modeling, and several other tasks that we will study in upcoming lectures. Notice that in most modern applications, data tends to be big — high-dimensional (large D), and with a large number of samples (large N). Without efficient computer processing techniques (e.g., distributed systems), many data science tasks would be impossible.