CHAPTER 3

A Brief Refresher on Working with Data

Data is central to every Intelligent System. This chapter gives a conceptual overview of working with data, introducing key concepts from data science, statistics, and machine learning. The goal is to establish a baseline of understanding to facilitate discussions and decision making between all participants of an Intelligent System project.

This chapter will cover:

- Structured data.
- Asking questions of data.
- Data models.
- Conceptual machine learning.
- Some common pitfalls of working with data.

Structured Data

Data is changing the world, that's for sure.

But what is it? A bunch of pictures on a disk? All the term papers you wrote in high school? The batting averages of every baseball player who ever played?

Yeah, all of that is data. There is so much data—oceans of numbers, filling hard disk drives all over the planet with ones and zeros. In raw form, data is often referred to as unstructured, and it can be quite difficult to work with.

When we turn data into intelligence, we usually work with data that has some structure to it. That is, data that is broken up into units that describe entities or events.

G. Hulten, *Building Intelligent Systems*, https://doi.org/10.1007/978-1-4842-3432-7_3

For example, imagine working with data about people. Each unit of the data describes a single person by: their weight, their height, their gender, their eye color, that sort of stuff.

One convenient way to think about it is as a table in a spreadsheet (Figure 3-1). There is one row for each person and one column for each property of the person. Maybe the first column of each row contains a number, which is the corresponding person's weight. And maybe the third column of each row contains a number, which is the person's height. On and on.

Weight	Gender	Height (Inches)	Eye Color	...
170	Male	70	Hazel	...
140	Female	60	Brown	...
60	Male	50	Blue	...
...	...	...	...	...

Figure 3-1. *An example of structured data*

The columns will usually contain numbers (like height and weight), or be chosen from a small number of categories (like brown or blue for eye color), or be short text strings like names. There are more advanced options, of course, but this simple scheme is quite powerful and can be used to represent all sorts of things (and unlock our ability to do statistics and machine learning on them).

For example:

- You can represent a web page by: the number of words it contains, the number of images it has in it, the number of links it contains, and the number of times it contains each of 1,000 common keywords.

- You can represent a visit to a search engine by: the query the user typed, the amount of time the user remained on the results page, the number of search results the user clicked, and the URL of the final result they clicked.
- You can represent a toasting event on our Internet toaster by: the temperature of the item placed in the toaster, the intensity setting the user selected, the number of times the user stopped and started the toaster before removing the item, and the total amount of time between when the item was placed in the toaster and when it was taken out.

These are simple examples. In practice, even simple concepts tend to have dozens or hundreds of elements in their representation (dozens or hundreds of columns per row). Choosing the right data sets to collect and the right representations to give them is critical to getting good results from data. And you'll probably get it wrong a few times before you get it right—be prepared to evolve.

Asking Simple Questions of Data

So what can you do with data? You can *ask it questions*. For example, in a data set of people you might want to know:

- What is the average height?
- Who is the tallest person?
- How many people are shorter than 5'5"?

These are pretty easy to answer; simply look through the data and calculate the values (sum up the numbers, or count the rows that meet the criteria).

In an Intelligent System you'll be asking plenty of similar questions. Things like:

- How many times per day do users take a particular action?
- What percentage of users click an ad after they engage with the intelligent experience?
- What's the average revenue per customer per month?

Answering questions like these is important to understanding if a system is meeting its goals (and making sure it isn't going haywire).

Another thing you can do with data is to *make projections*. Imagine a data set with a hundred people in it. Want to know the average height? No problem; just calculate it. But what if you want to know the height of the next person who will be added to the data set? Or whether the next person added to the set will be shorter or taller than 5'?

Can you do these types of things? Sure! Well, sort of, using basic statistics.

With a few simple assumptions, statistics can estimate the most likely height for the next person added to the data set. But statistics can do more. It can express exactly how accurate the estimate is; for example:

> *The most likely height of the next person is 5'10", and with 95% confidence the next person will be between 5'8" and 6' tall.*

This is called a confidence interval. The width of the confidence interval depends on how much data you have, and how "well behaved" the data is. More data results in a narrower window (for example, that the next person is 95% likely to be between 5'9" and 5'11"). Less data results in a wider one (like 5'5" to 6'3"). Why does this matter?

Let's imagine optimizing an Internet business by reducing customer support capacity to save money. Say the current system has capacity for 200 customer support calls per week. So looking into historical telemetry, you can calculate that the average week had 75 support calls and that the maximum number of calls in any week was 98. Intuitively, 200 is much higher than 75—the system must be wasting a lot of money. Cutting capacity in half (down to 100) seems safe, particularly because you don't have any week on record with a higher call volume than that.

But check the confidence interval. What if it came back that the most likely call volume is 75 per week, and with 95% confidence the next week will have between 40 and 110 calls? Well, then 100 doesn't seem like such an obviously good answer. In a data-intensive project, you should always ask for answers. But you should also ask how sure the answers are. And you should make sure the decisions you make take both the answer and the degree of certainty into account.

Working with Data Models

Models capture the answers from data, converting them into *more convenient* or *more useful* formats.

Technically, the projection we discussed previously was a model. Recall that the data set on call center volumes was converted into an average weekly volume of 75 with a 95% confidence interval of 40 - 110. The raw data might have been huge. It might have contained ten terabytes of telemetry going back a decade. But the model contained just four numbers: 75, 95%, 40, and 110.

And this simple model was more useful for making decisions about capacity planning, because it contained exactly the relevant information, and captured it in an intuitive way for the task at hand.

Models can also *fill in gaps in data* and estimate answers to questions that the data doesn't contain.

Consider the data set of people and their heights and weights. Imagine the data set contains a person who is 5'8" and a person who is 5'10", but no one who is 5'9".

What if you need to predict how much a 5'9" person will weigh?

Well, you could build a simple model. Looking into the data, you might notice that the people in the data set tend to weigh about 2.5 pounds per inch of height. Using that model, a 5'9" person would weigh 172.5 pounds. Great. Or is it?

One common way to evaluate the quality of models is called *generalization error*. This is done by reserving a part of the data set, hiding it from the person doing the modeling, and then testing the model on the holdout data to see how well the model does—how well it generalizes to data it has not seen.

For example, maybe there really was a 5'9" person in the data set, but someone hid them from the modeling process. Maybe the hidden 5'9"er weighed 150 pounds. But the model had predicted 172.5 pounds. That's 22.5 pounds off.

Good? Bad? Depends on what you need the model for.

If you sum up these types of errors over dozens or hundreds of people who were held out from the modeling process, you can get some sense of how good the model is. There are lots of technical ways to sum up errors and communicate the quality of a model, two important ones are:

- **Regression Errors** are the difference between a numeric value and the predicted value, as in the weight example. In the previous example, the model's regression error was 22.5.

- **Classification Errors** are for models that predict categories, for example one that tries to predict if a person is a male or female based on height and weight (and, yeah, I agree that model wouldn't have many friends). One way to measure classification errors involves accuracy: the model is 85% accurate at telling males from females.

Conceptual Machine Learning

Simple models can be useful. The weight-at-height model is very simple, and it's easy to find flaws with it. For example, it doesn't take gender into account, or body fat percent, or waist circumference, or where the person is from. Statistics are useful for making projections. Create a simple model, use it to answer questions. But models can be complex too—very, very, very complex.

Machine learning uses computers to improve the process of producing (complex) models from data. And these models can make projections from the data, sometimes surprisingly accurate—and sometimes surprisingly inaccurate.

A machine-learning algorithm explores the data to determine the best way to combine the information contained in the representation (the columns in the data set) into a model that generalizes accurately to data you haven't already seen.

For example, a machine-learning algorithm might predict any of the following:

- Gender by combining height, weight, age, and name.
- Weight using gender, height, age, and name.
- Height from the gender and weight.
- And so on…

And the way the model works can be very complicated, multiplying and scaling the various inputs in ways that make no sense to humans but produce accurate results.

There are probably thousands of machine-learning algorithms that can produce models of data. Some are fast; others run for days or weeks before producing a model. Some produce very simple models; some produce models that take megabytes or gigabytes of disk space. Some produce models that can make predictions very quickly on new data, a millisecond or less; some produce models that are computationally intensive to execute. And some do well on certain types of data, but fail at other types of problems.

There doesn't seem to be one universally best machine-learning algorithm to use for all situations—most work well in some situations, but poorly in others. Every few years a new technique emerges that does surprisingly well at solving important problems, and gets a lot of attention.

And some machine-learning people really like the algorithm they are most familiar with and will fight to the death to defend it despite any evidence to the contrary. You have been warned.

The machine-learning process generally breaks down into the following phases, all of which are difficult and require considerable effort from experts:

- **Getting the data to model.** This involves dealing with noise, and figuring out exactly what will be predicted and how to get the training examples for the modeling algorithm.
- **Feature engineering.** This involves processing the data into a data set with the right representation (columns in the spreadsheet) to expose to machine learning algorithms.
- **The modeling.** This involves running one or more machine-learning algorithms on the data set, looking at the mistakes they make, tweaking and tuning things, and repeating until the model is accurate enough.
- **The deployment.** This involves choosing which model to run, how to connect it into the system to create positive impact and minimal damage from mistakes.
- **The maintenance.** This involves monitoring that the model is behaving as expected, and fixing it if it starts to go out of control. For example, by rerunning the training algorithm on new data.

This book is about how to do these steps for large, complex systems.

Common Pitfalls of Working with Data

Data can be complicated, and there are a lot of things that can go wrong. Here are a few common pitfalls to keep in mind when working with data intensive systems, like Intelligent Systems.

Confidence intervals can be broken. Confidence intervals are very useful. Knowing that something is 95% likely to be between two values is great. But a 95% chance of being within an interval means there is a 5% chance of being outside the interval, too. And if it's out, it can be way out. What does that mean? 5% is one in twenty. So if you are estimating something weekly, for example to adjust capacity of a call center, one out of every twenty weeks will be out of interval—that's 2.6 times per year that you'll have too much capacity or not enough. 5% sounds small, like something that might never happen, but keep in mind that even low probability outcomes will happen eventually, if you push your luck long enough.

There is noise in your data. Noise is another word for errors. Like maybe there is a person in the dataset who has a height of 1", or 15'7". Is this real? Probably not. Where does the noise come from? All sorts of places. For example, software has bugs; data has bugs too. Telemetry systems log incorrect values. Data gets corrupted while being processed. The code that implements statistical queries has mistakes. Users turn off their computers at odd times, resulting in odd client states and crazy telemetry. Sometimes when a client has an error, it won't generate a data element. Sometimes it will generate a partial one. Sometimes this noise is caused by computers behaving in ways they shouldn't. Sometimes it is caused by miscommunication between people. Every large data set will have noise, which will inject some error into the things created from the data.

Your data has bias. Bias happens when data is collected in ways that are systematically different from the way the data is used. For example, maybe the people dataset we used to estimate height was created by interviewing random people on the streets of New York. Would a model of this data be accurate for estimating the heights of people in Japan? In Guatemala? In Timbuktu?

Bias can make data less useful and bias can come from all sorts of innocent places, like simple oversights about where the data is collected, or the context of when the data was collected. For example, users who "sorta-liked" something are less likely to respond to a survey than users who loved it or who hated it. Make sure data is being used to do the thing it was meant to do, or make sure you spend time to understand the implications of the bias your data has.

Your data is out of date. Most (simple) statistical and machine learning techniques have a big underlying assumption—that is, that things don't change. But things do change. Imagine building a model of support call volume, then using it to try to estimate support call volume for the week after a major new feature is released. Or the week

after a major storm. One of the main reasons to implement all the parts of an Intelligent System is to make the system robust to change.

You want the data to say things it doesn't. Sometimes data is inconclusive, but people like to have answers. It's human nature. People might downplay the degree of uncertainty by saying things like "the answer is 42" instead of saying "we can't answer that question" or "the answer is between 12 and 72." It makes people feel smarter to be precise. It can almost seem more polite to give people answers they can work with (instead of giving them a long, partial answer). It is fun to find stories in the data, like "our best product is the toothpaste, because we redid the display-case last month." These stories are so seductive that people will find them even where they don't exist.

Tip When working with data, always ask a few questions: Is this right? How sure are we? Is there another interpretation? How can we know which is correct?

Your models of data will make mistakes. Intelligent Systems, especially ones built by modeling data, are wrong. A lot. Sometimes these mistakes make sense. But sometimes they are totally counter-intuitive gibberish. And sometimes it is very hard to fix one mistake without introducing new ones. But this is fine. Don't be afraid of mistakes. Just keep in mind: any system based on models is going to need a plan for dealing with mistakes.

Summary

Data is changing the world. Data is most useful when it is structured in a way that lines up with what it needs to be used for. Structured data is like a spreadsheet, with one row per thing (person, event, web page, and so on), and one column per property of that thing (height, weight, gender).

Data can be used to answer questions and make projections. Statistics can answer simple questions and give you guidance on how accurate the answer is. Machine learning can create very complicated models to make very sophisticated projections. Machine learning has many, many useful tools that can help make accurate models of data. More are being developed all the time.

And data can be misused badly—you have to be careful.

For Thought

After reading this chapter, you should:

- Understand the types of things data is used for.
- Have some intuition about how to apply the things data can do.
- Have some intuition about how data can go wrong and lead to bad outcomes.

You should be able to answer questions like this:

- What is the biggest mistake you know of that was probably made because someone misused data?